Rational Suite®

COM Client API Reference

Rational Suite Extensibility

VERSION: 2002.05.00

PART NUMBER: 800-025147-000

WINDOWS

Rational
the software development company

support@rational.com
http://www.rational.com

Part Number: 800-025147-000

Version Number: 2002.05.00

Contents

Preface

This manual presents the COM client interfaces for developing applications with Rational Suite Extensibility (RSE). RSE delivers a comprehensive set of application programming interfaces (APIs) that provide a single platform on which to develop client and server capabilities between integrated products in the Rational Suite.

Audience

This guide is intended for administrators, project managers, and all members of the software development team, including requirements developers, software architects and developers, and quality engineers.

Other Resources

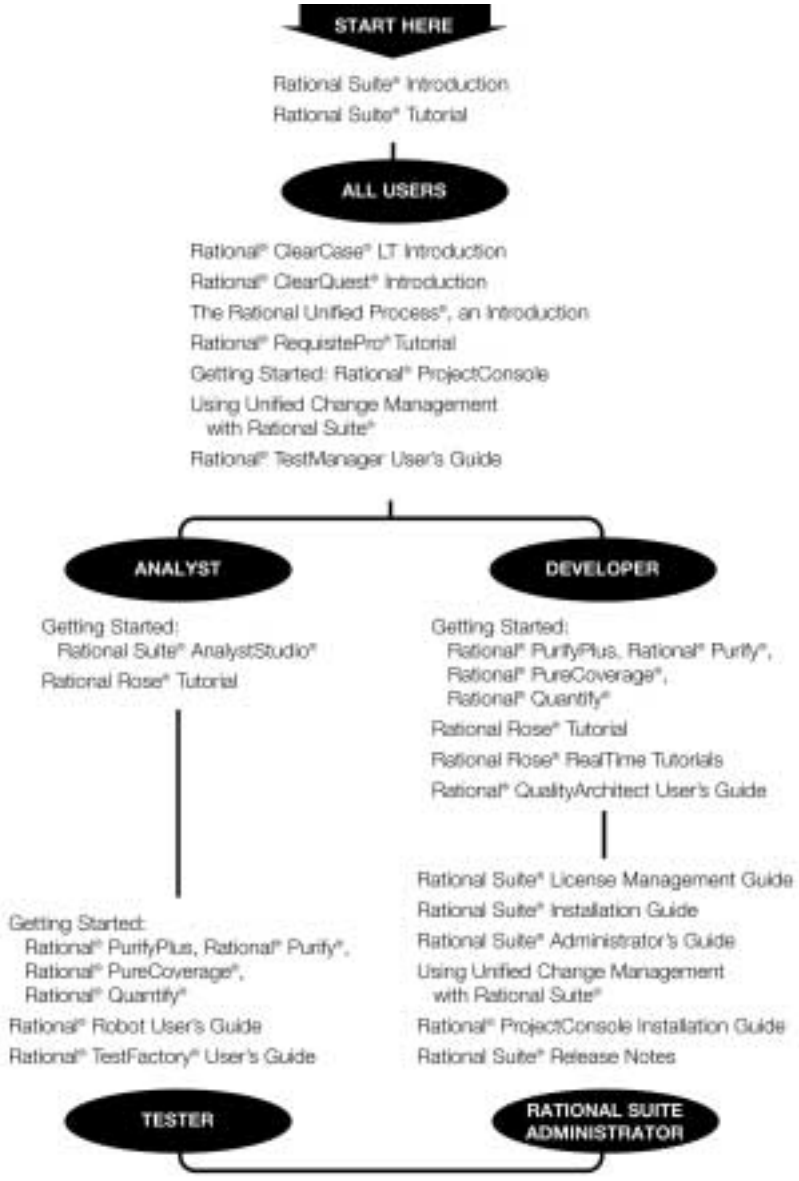
- Other RSE documentation:
  - Programmer's Guide to Application Development
  - Adapters Reference
  - Programmer's Guide to Adapter Development
- Rational extensibility API references:
  - ClearCase Reference Manual
  - ClearQuest API Reference
  - RequisitePro Extensibility Interface Online Help

    RequisitePro extensibility information is documented in the RequisitePro online help for the RequisitePro Extensibility Interface. It is available from the Help menu on the ReqPro tool palette.
  - Rose Extensibility Reference
  - Team Manager Extensibility Reference
- Online Help is available for Rational Suite.

  From a Suite tool, select an option from the **Help** menu.

- All manuals are available online, either in HTML or PDF format. The online manuals are on the Rational Solutions for Windows Online Documentation CD.
- To send feedback about documentation for Rational products, please send e-mail to techpubs@rational.com.
- For more information about Rational Software technical publications, see: http://www.rational.com/documentation.
- For more information on training opportunities, see the Rational University Web site: http://www.rational.com/university.

Rational Suite Documentation Roadmap



Contacting Rational Technical Support

If you have questions about installing, using, or maintaining this product, contact Rational Technical Support as follows:

Your Location	Telephone	Facsimile	E-mail
North America	(800) 433-5444 (toll free) (408) 863-4000 Cupertino, CA	(781) 676-2460 Lexington, MA	support@rational.com
Europe, Middle East, Africa	+31 (0) 20-4546-200 Netherlands	+31 (0) 20-4546-201 Netherlands	support@europe.rational.com
Asia Pacific	+61-2-9419-0111 Australia	+61-2-9419-0123 Australia	support@apac.rational.com

Note: When you contact Rational Technical Support, please be prepared to supply the following information:

- Your name, company name, telephone number, and e-mail address
- Your operating system, version number, and any service packs or patches you have applied
- Product name and release number
- Your case ID number (if you are following up on a previously reported problem)

COM Client Interfaces

This section lists the available client interfaces contained in the RDSI core library (RDSICore.dll). To view this library:

- In Visual Basic, select View/Object Browser to view the RDSI core library.
- In Visual C++, include a `#import RDSICore.dll` statement to generate temporary code that gets compiled for the C++ object oriented version of this API.

This manual lists the interfaces alphabetically. Each interface includes a table listing its methods. Following the table, the methods are listed alphabetically and include:

- Description
- Argument types, if any
- Argument descriptions
- Code example

There are also additional selected VB script code examples (for example, creating a session with a VB script).

Overview

A client application starts with creating an IRDSISession object. Creating a session object returns all available adapters (IAdapterCollection). Given an adapter, you can receive the artifact types available through that adapter .

The client interfaces include:

- Session interface
- Adapter interfaces
- Artifact interfaces
- Locator interfaces
- Property interfaces
- Relationship interfaces
- Query interfaces

- Method interfaces
- Graphics support interfaces

Table 1 lists the actual interfaces for these types. *Appendix A* provides class diagrams for these interfaces.

Table 1 Client Interfaces

Type	Name
Session Interface	IRDSISession
Adapter Interfaces	IAdapter
	IAdapterCollection
Artifact Interfaces	IArtifact
	IArtifactArgument
	IArtifactArgumentCollection
	IArtifactCollection
	IArtifactIterator
	IArtifactPersist
	IArtifactType
	IArtifactTypeCollection
Locator Interfaces	IArtifactLocator
	IArtifactLocatorCollection
	IArtifactLocatorType
	IArtifactLocatorTypeCollection
Property Interfaces	IArtifactProperty
	IArtifactPropertyCollection
	IArtifactPropertyType
	IArtifactPropertyTypeCollection
Relationship Interfaces	IRelationshipType
	IRelationshipTypeCollection
Query Interfaces	IArtifactFilter

Type	Name
	IArtifactFilterCollection
	IFilterComparison
	IFilterComparisonCollection
Method Interfaces	IMethod
	IMethodType
	IMethodTypeCollection
Graphics Support Interfaces	IArtifactGraphicsFormatType
	IArtifactGraphicsFormatTypeCollection
	IArtifactGUI

IAdapter Interface

This is the default interface for the RSE Adapter object. This object encapsulates information about the underlying provider.

Adapter Properties

Table 2 lists the IAdapter properties.

Table 2 IAdapter Properties

Property	Description
Name	Returns the name of the adapter
StaticArtifactTypes	Returns the top-level static artifact types registered by this adapter.
VersionString	Returns the provider version string

Adapter.Name Property

Returns the name of the adapter. This is a read-only property.

Name

Return Value

Returns the name of the adapter.

Arguments

None

Example

```
Dim theRDSISession2 As RDSISession
Dim theAdapter2 As Adapter
Dim theAdapter2Name As String
Dim theArtifact2 As Artifact
Dim theArtifactTypes2 As ArtifactTypeCollection
Dim pInternalObject2 As Variant
```

```
// GetRDSISession is a user defined function that creates a Session.
Set theRDSISession2 = GetRDSISession()
// GetCurrentAdapter is a user defined function that retrieves
// an instance of an adapter.
Set theAdapter2 = GetCurrentAdapter(1, theRDSISession2)
theAdapter2Name = theAdapter2.Name
```

Adapter.StaticArtifactTypes

Returns the collection of the top-level static artifact types that are registered by this adapter. This returns one project level type.

StaticArtifactTypes

Return Value

Returns the ArtifactTypeCollection object (IArtifactTypeCollection interface).

Arguments

None

Example

```
    Dim theRDSISession As RDSISession
    Dim theAdapter As Adapter
    Dim theArtifactTypes As ArtifactTypeCollection
    Dim theArtifactType As ArtifactType
// GetRDSISession is a user defined function that creates a Session.
  Set theRDSISession = GetRDSISession()
// GetCurrentAdapter is a user defined function that retrieves
// an instance of an adapter.
  Set theAdapter = GetCurrentAdapter()
    Set theArtifactTypes = theAdapter.StaticArtifactTypes
```

Adapter.VersionString Property

Returns the version string of the provider. This is a read-only property.

VersionString

Return Value

Returns the version string of the provider.

Arguments

None

Example

```
Dim theRDSISession2 As RDSISession
Dim theAdapter2 As Adapter
Dim theAdapter2Name As String
Dim theArtifact2 As Artifact
// GetRDSISession is a user defined function that creates a Session.
Set theRDSISession2 = GetRDSISession()
// GetCurrentAdapter is a user defined function that retrieves
// an instance of an adapter.
Set theAdapter2 = GetCurrentAdapter(1, theRDSISession2)
theAdapter2Name = theAdapter2.VersionString
```

Adapter Methods

Table 3 lists the IAdapter properties.

Table 3 IAdapter Methods

Method	Description
GetStaticArtifactType_ID	Returns the ArtifactType corresponding to the ClassID.

Adapter.GetStaticArtifactType_ID

Returns the ArtifactType corresponding to the ClassID.

GetStaticArtifactType_ID (LONG ClassID)

Return Value

Returns the ArtifactType object (ArtifactType interface) corresponding to the ClassID.

Argument

ClassID

The unique id of a static artifact type.

Example

```
   Dim theRDSISession As RDSISession
   Dim theAdapter As Adapter
   Dim theArtifactType As ArtifactType
   Dim theClassID As Long
‘ GetRDSISession is a user defined function that creates a Session.
  Set theRDSISession = GetRDSISession()
‘ GetCurrentAdapter is a user defined function that retrieves
‘ an instance of an adapter.
  Set theAdapter = GetCurrentAdapter()
   Set theArtifactType = theAdapter.StaticArtifactType_ID(theClassID)
```

IAdapterCollection Interface

This interface encapsulates information about the list of available adapters.

You create an adapter collection, in VB script, as follows:

```
Dim theSession, theArtifact, theAdapterCollection,theTestAdapter
Set theTestAdapter = theSession.Adapters.Item(2)


' Creating the Adapter collection object
Set theAdapterCollection =
WScript.CreateObject("RDSICore.AdapterCollection.1")
```

Table 5 lists the IAdapterCollection methods.

AdapterCollection Properties

Table 4 lists the IAdapterCollection properties.

Table 4 AdapterCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

AdapterCollection.Count Method

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim theAdapterCollection52 As AdapterCollection
```

```
Dim Count5 As Integer
Count5 = theAdapterCollection52.Count
```

VB Script Example

```
Dim theSession, theAdapterCollection


' Creating the Adapter collection object
Set theAdapterCollection =
WScript.CreateObject("RDSICore.AdapterCollection.1")
Dim thecount
thecount = theAdapterCollection.Count
```

AdapterCollection.Item Method

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns an Adapter object (IAdapter interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theRDSISession5 As RDSISession
Dim theAdapter5 As Adapter
Dim theAdapterCollection51 As AdapterCollection
Dim theAdapterCollection52 As AdapterCollection
Dim Count5 As Integer
Dim IsModify5 As Boolean
Dim Index5 As Variant
Dim theAdapter51 As Adapter

Set theRDSISession5 = New RDSISession
Set theAdapter5 = theRDSISession5.Adapters.Item(2)
```

```
Set theAdapter51 = theRDSISession5.Adapters.Item(3)

For adapterid = 0 To theRDSISession5.Adapters.Count - 1
Set theAdapter5 = theRDSISession5.Adapters.Item(adapterid)
Set theAdapter51 = theAdapterCollection52.Item(adapterid)
Next adapterid
```

AdapterCollection Methods

Table 5 lists the IAdapterCollection methods.

Table 5 IAdapterCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
Count	Returns the count of items in the collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Item	Returns a reference to the object at the given index.
Remove	Removes the object at the specified index from the collection.

AdapterCollection.Add Method

Appends the specified object to the end of the collection.

Add (IAdapter)

Return Value

None

Argument

Adapter

An Adapter object (IAdapter interface).

Example

```
Dim theRDSISession5 As RDSISession
Dim theAdapter5 As Adapter
Dim theAdapterCollection51 As AdapterCollection
Dim theAdapterCollection52 As AdapterCollection
Dim Count5 As Integer
Dim theAdapter51 As Adapter

Set theRDSISession5 = New RDSISession
Set theAdapter5 = theRDSISession5.Adapters.Item(2)
Set theAdapter51 = theRDSISession5.Adapters.Item(3)

Set theAdapterCollection51 = New AdapterCollection
theAdapterCollection51.Add theAdapter5
```

VB Script Example

```
Dim theSession, theArtifact, theAdapterCollection,theTestAdapter
Set theTestAdapter = theSession.Adapters.Item(2)

' Creating the Adapter collection object
Set theAdapterCollection =
WScript.CreateObject("RDSICore.AdapterCollection.1")
theAdapterCollection.Add theTestAdapter
```

AdapterCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IAdapterCollection)

Return Value

None

Argument

Collection

An AdapterCollection object (IAdapterCollection interface).

Example

```
Dim theRDSISession5 As RDSISession
Dim theAdapter5 As Adapter
Dim theAdapterCollection51 As AdapterCollection
Dim theAdapterCollection52 As AdapterCollection
Dim Count5 As Integer
Dim IsModify5 As Boolean
Dim Index5 As Variant
Dim theAdapter51 As Adapter

Set theRDSISession5 = New RDSISession
Set theAdapter5 = theRDSISession5.Adapters.Item(2)
Set theAdapter51 = theRDSISession5.Adapters.Item(3)
' Testing the AddCollection method for the AdapterCollection Interface
Set theAdapterCollection52 = New AdapterCollection
theAdapterCollection52.AddCollection theAdapterCollection51
```

AdapterCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theAdapterCollection52 As AdapterCollection
Dim IsModify5 As Boolean
IsModify5 = theAdapterCollection52.IsModifiable
```

AdapterCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim theRDSISession5 As RDSISession
Dim theAdapter5 As Adapter
Dim theAdapterCollection51 As AdapterCollection
Dim theAdapterCollection52 As AdapterCollection
Dim Count5 As Integer
Dim IsModify5 As Boolean
Dim Index5 As Variant
Dim theAdapter51 As Adapter

Set theRDSISession5 = New RDSISession
Set theAdapter5 = theRDSISession5.Adapters.Item(2)
Set theAdapter51 = theRDSISession5.Adapters.Item(3)
Set theAdapterCollection51 = New AdapterCollection
‘Add an adapter
theAdapterCollection51.Add theAdapter5
‘Remove the adapter
theAdapterCollection51.Remove theAdapter5
```

IArtifact Interface

IArtifact interface is the default interface of the Artifact object. Use this interface to gain access to all data associated with the artifact.

This class includes methods for:

- Related objects
  - Adapter
  - Parent
  - RegisteredTypes
  - Type
- Artifact locators
  - CreateLocator
  - GetDefaultArtifactID
  - GetInternalObject
  - Persist
- Property support
  - Description
  - GetPropertyValue
  - Name
  - Properties
  - SetPropertyValue
- Related artifact support
  - CreateArtifact
  - DeleteArtifact
  - GetRelatedArtifacts
- Graphics support
  - GUI
  - RenderToFile
  - RenderToFileEx

Artifact Properties

Table 6 lists the Artifact properties.

Table 6 Artifact Properties

Property	Description
Adapter	Returns the adapter that implements this artifact.
Description	Returns a description of the artifact.
GUI	Returns a handle to the IArtifactGUI interface associated with this object if it is supported.
Name	Returns the name of the artifact.
Parent	Returns the parent associated with the artifact. If the artifact has no parent, the return value is NULL.
Persist	Returns a handle to the IArtifactPersist interface associated with this object if it is supported.
Properties	Returns the collection of all properties associated with this Artifact.
RegisteredTypes	Returns the collection of all Artifact Types that exist within the scope of this artifact.
Type	Returns the ArtifactType object associated with this Artifact. This call should normally succeed and return a valid object.

Artifact.Adapter Property

Returns the adapter that implements this artifact.

Adapter

Return Value

Returns an Adapter object (IAdapter interface).

Arguments

None

Example

```
Dim theRDSISession6 As RDSISession
Dim theAdapter6 As Adapter
```

```
Dim retAdapter6 As Adapter
Dim theArtifact6 As Artifact
Dim retAdapterStr6 As String
Dim filepath16 As String
Dim filepath6 As String
Dim lretval6 As String
Dim hKeytype6 As Long
Dim ipString6 As String
Dim hKey6 As Long
Dim valString6 As String
Dim regvalString6 As String
Dim theArtifactType6 As ArtifactType
Dim theArtifactLocator6 As ArtifactLocator


hKeytype6 = HKEY_LOCAL_MACHINE
ipString6 = "SOFTWARE\Rational Software\RequisitePro\4.0"
lretval6 = RegOpenKeyEx(hKeytype6, ipString6, 0, KEY_ALL_ACCESS,
hKey6)
valString6 = "InstallDirectory"
lretval6 = QueryValueEx(hKey6, valString6, regvalString6)
filepath6 = "ReqPro|Project(Path='" + regvalString6 +
"\samples\Learning_Project\Business\Learning_Business.RQS')"
Set theRDSISession6 = New RDSISession
Set theArtifact6 = theRDSISession6.LocateArtifact(filepath6)
retAdapterStr6 = theArtifact6.Adapter.Name
```

Artifact.Description Property

Gets the artifact Description property.

Description

Return Value

Returns the Description property of the Artifact.

Arguments

None

Example

```
Dim theArtifactDescription6 As String
Dim theArtifactType6 As ArtifactType
theArtifactDescription6 = theArtifact6.Description()
```

Artifact.GUI Property

Returns a handle to the IArtifactGUI interface associated with this object if it is supported.

GUI

Return Value

Returns an ArtifactGUI object (IArtifactGUI interface).

Arguments

None

Example

```
Dim theArtifactGUI6 As Boolean
theArtifactGUI6 = theArtifact6.GUI.CanShow
```

Artifact.Name Property

Get Name property. Returns the name of the artifact.

Name

Return Value

Returns the BSTR name property

Arguments

None

Example

```
Dim theArtifact As Artifact
Dim theArtifactname As String
theArtifactname = theArtifact.Name
```

Artifact.Parent Property

Returns the parent associated with the artifact. If the artifact has no parent, the return value is NULL.

Given an instance of an object (artifact), it is contained within a parent, if it has a parent object.

Parent

Return Value

Returns an Artifact object (IArtifact interface).

Arguments

None

Example

```
Dim theArtifact As Artifact
Dim theParentArtifact As Artifact
Set theParentArtifact = theArtifact.Parent
```

Artifact.Persist Property

Offered as a convenience for Visual Basic programmers instead of QueryInterface, this method returns a handle to the ArtifactPersist interface associated with this object if it is supported, otherwise NULL.

Persist

Return Value

Returns an ArtifactPersist object (IArtifactPersist interface).

Arguments

None

Example

```
Dim theArtifactPersist As Boolean
Dim theArtifact As Artifact
theArtifactPersist = theArtifact6.Persist.CanSave
```

Artifact.Properties Property

Returns the collection of all properties associated with this Artifact

Properties

Return Value

Returns an ArtifactPropertyCollection object (IArtifactPropertyCollection interface).

Arguments

None

Example

```
Dim theArtifact As Artifact
Dim theArtifactProperties As ArtifactPropertyCollection
Set theArtifactProperties = theArtifact.Properties
```

Artifact.RegisteredTypes Property

Returns the collection of all Artifact Types that exist within the scope of this artifact. This includes the static types registered by the adapter and any dynamic types registered by this or any parent artifact.

RegisteredTypes

Return Value

Returns the ArtifactTypeCollection object (IArtifactTypeCollection interface).

Arguments

None

Example

```
Dim theArtifact As Artifact
Dim theArtifactRegisteredTypes As ArtifactTypeCollection
Set theArtifactRegisteredTypes = theArtifact.RegisteredTypes
```

Artifact.Type Property

Returns the ArtifactType object associated with this Artifact.

Type

Return Value

Returns the ArtifactType object (IArtifactType interface).

Arguments

None

Example

```
Dim theArtifactType61 As ArtifactType
Dim theArtifact6 As Artifact
Set theArtifactType61 = theArtifact6.Type
```

Artifact Methods

Table 7 lists the Artifact methods.

Table 7 Artifact Methods

Method	Description
CreateArtifact	Creates a new child artifact object, given a relationship type.
CreateArtifact_ID	Creates a new child artifact object of the specified type id.
CreateLocator	Returns the default locator for the given artifact type.
CreateLocatorEx	Creates a locator for the artifact using the specific locator type.
CreateLocatorEx_ID	Returns a locator for the given artifact type, given a specific locator type id.
CreateMethod	Creates a method based on method type.
CreateMethod_ID	Creates a method based on method type id.
DeleteArtifact	Deletes the artifact object.
DeleteArtifact_ID	Deletes the artifact object of the specified type id.
GetDefaultArtifactID	Returns the default artifact id for the given artifact.
GetInternalObject	Returns the IUnknown of the internal object associated with the artifact.
GetPropertyValue	Returns a specific property value by property type.

Method	Description
GetPropertyValue_ID	Returns a specific property value by property id.
GetRelatedArtifacts	Returns a collection of artifacts of a given relationship type.
GetRelatedArtifacts_ID	Returns a collection of artifacts of a given relationship type id.
RenderToFile	Creates a file containing the rendered image of the artifact.
RenderToFileEx	Creates a file containing the rendered image of the artifact.
SetPropertyValue	Sets a specific property value by property type.
SetPropertyValue_ID	Sets a specific property value by property id.

Artifact.CreateArtifact Method

Creates a new artifact object, given a RelationshipType. This method creates the ArtifactType that is associated to your given Artifact by the specified RelationshipType. The return value is a reference to the created ArtifactType object.

Note: Not all Artifacts may be created. Before calling CreateArtifact, call pType->CreateAndDeleteAllowed ().

CreateArtifact (RelationshipType)

Return Value

Returns the created Artifact object (IArtifact interface).

Arguments

RelationshipType

A reference to the relationship (IRelationshipType interface) that defines how the artifact is related to this one.

Example

```
Dim theArtifact As Artifact
Dim theNewArtifact As Artifact
Dim theRelationshipType As RelationshipType
Set theNewArtifact = theArtifact.CreateArtifact theRelationshipType
```

Artifact.CreateArtifact_ID Method

CreateArtifact_ID is intended for custom purposes. See CreateArtifact.

This method creates a new child artifact object of the specified type id, given a relationship type id.

Note: Not all Artifacts may be created. Before calling CreateArtifact_ID, call CreateAndDeleteAllowed ().

CreateArtifact_ID (LONG ClassID, LONG RelationshipTypeID)

Return Value

Returns the created Artifact object (IArtifact interface).

Arguments

ClassID

A unique id for the ArtifactType object that describes the type of artifact to create. This can be obtained given an artifact type name using Type.ChildTypes.Item (TypeName).

RelationshipTypeID

A unique id for the relationship that defines how the artifact is related to this one.

Example

```
Dim theArtifact As Artifact
Dim theArtifactType As ArtifactType
Dim theNewArtifact As Artifact
‘Get the ClassID
Dim theClassID As Long
theClassID = theArtifactType.ClassID
‘Get theRelationshipTypeID
Dim theRelationshipTypeID As Long
Dim the RelationshipType As RelationshipType
theRelationshipTypeID = theRelationshipType.RelationshipTypeID
‘Create new artifact
Set theNewArtifact = theArtifact.CreateArtifact_ID(theClassID,
theRelationshipTypeID)
```

Artifact.CreateLocator Method

Returns the default locator for the given artifact type.

CreateLocator (eLocatorType LocatorType)

Return Value

Returns an ArtifactLocator object (IArtifactLocator interface).

Argument

LocatorType

An optional argument for specifying the locator type to create.

Example

```
Dim theArtifact30 As Artifact
Dim theLocator30 As ArtifactLocator
Dim theArtifactType30 As ArtifactType
Set theArtifactType30 = theArtifact30.Type
Set theLocator30 =
theArtifactType30.CreateLocator(rsLocator_Display_Name)
```

Artifact.CreateLocatorEx Method

Creates a locator for the artifact using the specific locator type. The locator is initialized with the argument values specific to the artifact. If a relative locator is passed, then this locator is used. However, the artifact type of the relative locator must match the relative artifact type of theLocatorType. If no relative locator is passed and one is needed, then an appropriate default locator for the relative type is chosen.

CreateLocatorEx (IArtifactLocatorType LocatorType, IArtifactLocator RelativeLocator)

Return Value

Returns an ArtifactLocator object (IArtifactLocator interface).

Arguments

LocatorType

Specifies the locator type to create.

RelativeLocator

An optional argument for specifying a relative locator. The ArtifactType of the RelativeLocator must match relative artifact type of the LocatorType.

Example

```
Dim theArtifact As Artifact
Dim theLocator As ArtifactLocator
Dim theLocatorType As ArtifactLocatorType
Set theLocator = theArtifact.CreateLocatorEx(theLocatorType)
```

Artifact.CreateLocatorEx_ID Method

CreateLocatorEx_ID is intended for custom purposes. See CreateLocatorEx.

Creates a locator using the specific locator type id. Otherwise, identical to CreateLocatorEx.

Creates a locator for the artifact using the specific locator type. The locator is initialized with the argument values specific to the artifact. If a relative locator is passed, then this locator is used. However, the artifact type of the relative locator must match the relative artifact type of theLocatorType. If no relative locator is passed and one is needed, then an appropriate default locator for the relative type is chosen.

CreateLocatorEx_ID (LONG ClassID, Int ArtifactLocatorTypeID, IArtifactLocator RelativeLocator)

Return Value

Returns an ArtifactLocator object (IArtifactLocator interface).

Arguments

ClassID

A unique id for the ArtifactType of the locator to create.

LocatorType

Specifies the unique id of the locator type to create.

RelativeLocator

An optional argument for specifying a relative locator. The ArtifactType of the RelativeLocator must match relative artifact type of the LocatorType.

Example

```
Dim theArtifact As Artifact
Dim theArtifactType As ArtifactType
Dim theLocator As ArtifactLocator
```

```
Dim theLocatorType As ArtifactLocatorType
Set theArtifactType = theArtifact.Type
‘Get the ClassID
Dim theClassID As Long
theClassID = theArtifactType.ClassID
‘Get the LocatorTypeID
Dim theLocatorTypeID As String
theLocatorTypeID = theLocatorType.LocatorTypeID
Set theLocator = theArtifact.CreateLocatorEx_ID(theClassID,
theLocatorTypeID)
```

Artifact.CreateMethod Method

Creates a method based on MethodType.

CreateMethod (IMethodType MethodType)

Return Value

Returns a reference to a Method object (IMethod interface).

Argument

MethodType

The type of method to create.

Example

```
Dim theArtifact As Artifact
Dim theMethod As Method
Dim theMethodType As MethodType
theMethod = theArtifact.CreateMethod(theMethodType)
```

Artifact.CreateMethod_ID Method

CreateMethod_ID is intended for custom purposes. See CreateMethod.

Creates a method based on the artifact type and method id.

CreateMethod_ID (LONG ClassID, LONG MethodID)

Return Value

Returns a reference to a Method object (IMethod interface).

Arguments

ClassID

The unique id of the ArtifactType..

MethodID

The unique id of the type of method to create.

Example

```
Dim theArtifact As Artifact
Dim theMethod As Method
‘Get the ClassID
Dim theClassID As Long
Dim theArtifactType As ArtifactType
theArtifactType = theArtifact.Type
theClassID = theArtifactType.ClassID
‘Get the MethodTypeID
Dim theMethodID As String
Dim theMethodType As MethodType
theMethodID = theMethodType.MethodTypeID
theMethod = theArtifact.CreateMethod_ID(theClassID, theMethodID)
```

Artifact.DeleteArtifact Method

Deletes the Artifact object. After being deleted, the object becomes inaccessible. For clients to know how to delete an arbitrary artifact, they need specific knowledge of the relationship type needed for that artifact.

Note: Not all Artifacts may be deleted. Before calling Delete, call pType->CreateAndDeleteAllowed().

DeleteArtifact (IRelationshipType RelationshipType, IArtifact ArtifactToDelete

Return Value

None

Arguments

RelationshipType

A reference to the relationship that defines how the artifact to be deleted is related to this one.

ArtifactToDelete

Reference to the artifact to be deleted.

Example

```
Dim the RelationshipType As RelationshipType
Dim theArtifact As Artifact
Dim theArtifactToDelete As Artifact
theArtifact.DeleteArtifact(theRelationshipType, theArtifactToDelete)
```

Artifact.DeleteArtifact_ID Method

DeleteArtifact_ID is intended for custom purposes. See DeleteArtifact.

Deletes the Artifact object of the specified type id. After being deleted, the object becomes inaccessible. For clients to know how to delete an arbitrary artifact, they need specific knowledge of the relationship type needed for that artifact.

Note: Not all Artifacts may be deleted. Before calling Delete, call CreateAndDeleteAllowed().

DeleteArtifact_ID (LONG ClassID, LONG RelationshipTypeID, IArtifact ArtifactToDelete)

Return Value

None

Arguments

ClassID

A unique id of the ArtifactType of the locator to delete.

RelationshipTypeID

The unique id of the relationship that defines how the artifact to be deleted is related to this one.

ArtifactToDelete

Reference to the artifact to be deleted.

Example

```
Dim theArtifact As Artifact
Dim theArtifactToDelete As Artifact
```

```
‘Get the ClassID
Dim theClassID As Long
Dim theArtifactType As ArtifactType
theClassID = theArtifactType.ClassID
‘Get theRelationshipTypeID
Dim theRelationshipTypeID As Long
Dim the RelationshipType As RelationshipType
theRelationshipTypeID = theRelationshipType.RelationshipTypeID
theArtifact.DeleteArtifact_ID(theClassID, theRelationshipTypeID,
theArtifactToDelete)
```

Artifact.GetDefaultArtifactID Method

Returns the default artifact id for the given artifact. The namespace calls GetDefaultLocator using the Artifact's type. Using the locator, it calls GetArtifactID, passing pArtifact.

GetDefaultArtifactID (eLocatorType LocatorType)

Return Value

Returns the ArtifactID.

Argument

LocatorType

An optional argument for specifying the locator type.

Example

```
Dim theArtifact6 As Artifact
Dim theArtifactIDText As String
' Getting the Immutable Artifact ID from the Artifact
theArtifactIDText =
theArtifact6.GetDefaultArtifactID(rsLocator_Immutable_ID)
' Getting the Display Name Artifact ID from the Artifact
theArtifactIDText =
theArtifact6.GetDefaultArtifactID(rsLocator_Display_Name)
```

Artifact.GetInternalObject Method

This read-only property returns the IUnknown of the "internal object" associated with the artifact. This is generally a COM object defined by the specific provider. The caller must have prior knowledge of the capabilities and interfaces supported by this object. If the provider does not support a COM interface to its object, the method returns NULL.

GetInternalObject

Return Value

Returns the IUnknown of the internal object associated with the artifact.

Arguments

None

Example

```
Dim theArtifact6 As Artifact
Dim theObject6 As Variant
Set theObject6 = theArtifact6.GetInternalObject()
```

Artifact.GetPropertyValue Method

This method gets a specific property value by property type. Algorithms that get properties on large numbers of artifacts should use this mechanism instead of using the Property collection.

GetPropertyValue (IArtifactPropertyType)

Return Value

A variant that specifies the value to be returned. Must be consistent with the data type supported by the associated property type.

Argument

Type

A property type. This is an instance of a property type belonging to the Type member of this artifact. This value may be obtained given the name of the property using Type.PropertyTypes.Item (Name).

Example

```
Dim theArtifact6 As Artifact
Dim theArtifactPropertyType6 As ArtifactPropertyType
Dim theArtifactPropertyValue6 As String
For Prop = 0 To theArtifact6.Type.PropertyTypes.Count - 1
Set theArtifactPropertyType6 =
theArtifact6.Type.PropertyTypes.Item(Prop)
theArtifactPropertyValue6 =
CStr(theArtifact6.GetPropertyValue(theArtifactPropertyType6))
Next Prop
```

Artifact.GetPropertyValue_ID Method

GetPropertyValue_ID is intended for custom purposes. See GetPropertyValue.

This method gets a specific property value identified by the property type id. Works just like GetPropertyValue. Uses the ClassID of the ArtifactType and PropertyID of the property belonging to that type to specify the property.

GetPropertyValue_ID (LONG ClassID, LONG PropertyID)

Return Value

Returns a variant that specifies the value to be returned. Must be consistent with the data type supported by the associated property type.

Arguments

ClassID

A unique id of the ArtifactType of this artifact.

PropertyID

A property type unique id.

Example

```
Dim theArtifact As Artifact
Dim theArtifactPropertyType As ArtifactPropertyType
Dim theArtifactPropertyValue As String
'Get the ClassID
Dim theClassID As Long
Dim theArtifactType As ArtifactType
```

```
theClassID = theArtifactType.ClassID
'Get the PropertyID
Dim thePropertyID As Long
Dim theArtifactPropertyType As ArtifactPropertyType
thePropertyID = theArtifactPropertyType.PropertyID
theArtifactPropertyValue =
CStr(theArtifact.GetPropertyValue_ID(theClassID, thePropertyID))
```

Artifact.GetRelatedArtifacts Method

Returns a collection of Artifacts of a given relationship type. The relationship type of artifact to return is specified using a reference to a RelationshipType object. An empty collection is returned if this Artifact has no artifacts of the specified relationship type.

GetRelatedArtifacts (IRelationshipType)

Return Value

Returns a collection of Artifacts of a given relationship type (IArtifactCollection interface).

Arguments

RelationshipType

A reference to RelationshipType object that describes type of artifacts to be returned. This can be obtained, given a relationship type name using RelationshipType.GetRelatedArtifactType().

Filter

An optional argument that is an artifact filter object (IArtifactFilter interface).

Example

```
Dim theArtifact As Artifact
Dim theRelationship As RelationshipType
Dim theRelatedArtifacts As ArtifactCollection
Set theRelatedArtifacts =
theArtifact.GetRelatedArtifacts(theRelationship)
```

Artifact.GetRelatedArtifacts_ID Method

GetRelatedArtifacts_ID is intended for custom purposes. See GetRelatedArtifacts.

Returns a collection of Artifacts of a given relationship type. The relationship type of artifact to return is specified using a reference to an RelationshipType object.

GetRelatedArtifacts_ID (LONG ClassID, LONG RelationshipID, IArtifactFilter Filter)

Return Value

Returns a collection of Artifacts of a given relationship type (IArtifactCollection interface).

Arguments

ClassID

The unique id of the ArtifactType of this artifact.

RelationshipID

The unique id of the RelationshipType object that describes type of artifacts to be returned. This can be obtained, given an artifact type name using Type.GetRelatedArtifactType().

Filter

An optional argument that is an artifact filter object (IArtifactFilter interface).

Example

```
Dim theArtifact As Artifact
Dim theRelatedArtifacts As ArtifactCollection
‘Get the ClassID
Dim theClassID As Long
Dim theArtifactType As ArtifactType
theClassID = theArtifactType.ClassID
‘Get the RelationshipID
Dim theRelationshipID As Long
Dim theRelationship As RelationshipType
theRelationshipID = theRelationship.RelationshipID
Set theRelatedArtifacts =
theArtifact.GetRelatedArtifacts_ID(theClassID, theRelationshipID)
```

Artifact.RenderToFile Method

Creates a file containing the rendered image of the artifact, allowing it to be viewed. The format object can be retrieved using the ArtifactType.

Before you can call RenderToFile, you must see if this artifact type supports rendering graphics to files. Call ArtifactType.GraphicsFormats. This returns the collection of graphics formats for this artifact. If there are objects contained in this collection (if the GraphicsFomatTypeCollection.Count > 1), then you can render the artifact to a file.

RenderToFile (IArtifactGraphicsFormatType Format, BSTR FileName)

Return Value

None

Arguments

Format

An ArtifactGraphicsFormatType object (IArtifactGraphicsFormatType interface).

FileName

The name of the file.

Example

```
Dim GraphicsFormats As ArtifactGraphicsFormatTypeCollection
Dim GraphicsFormat As ArtifactGraphicsFormatType
Dim FormatID As Integer
Dim Results As String

Set GraphicsFormats = theArtifact.Type.GraphicsFormats
If GraphicsFormats.Count = 0 Then
   PrintComment 1, "This object does not support rendering graphics."
   Exit Function
End If
For FormatID = 0 To GraphicsFormats.Count - 1
   Set GraphicsFormat = GraphicsFormats.Item(FormatID)
   Dim Ext As String
   Ext = GraphicsFormat.Format
   theArtifact.RenderToFile GraphicsFormat, "File." + Ext
```

```
   Results = theArtifact.RenderToFileEx(GraphicsFormat,
"File(Path='FileEx." + Ext + "')|Format(Width=1 Height=1
UnitsPerInch=1)")
Next FormatID
```

Artifact.RenderToFileEx Method

Creates a file containing the rendered image of the artifact, allowing it to be viewed. The format object can be retrieved using the ArtifactType. Extended paramter information can be passed in the Parameters string. The expected format of the string is specific to each artifact implementation. Implementation specific information is also returned in the pResults string.

Before you can call RenderToFile, you must see if this artifact type supports rendering graphics to files. Call ArtifactType.GraphicsFormats. This returns the collection of graphics formats for this artifact. If there are objects contained in this collection (if the GraphicsFomatTypeCollection.Count > 1), then you can render the artifact to a file.

RenderToFileEx (IArtifactGraphicsFormatType Format, BSTR Parameters)

Return Value

Returns the results information.

Arguments

Format

The format type.

Parameters

A BSTR for extended parameter information.

Example

```
Dim GraphicsFormats As ArtifactGraphicsFormatTypeCollection
Dim GraphicsFormat As ArtifactGraphicsFormatType
Dim FormatID As Integer
Dim Results As String

Set GraphicsFormats = theArtifact.Type.GraphicsFormats
If GraphicsFormats.Count = 0 Then
   PrintComment 1, "This object does not support rendering graphics."
   Exit Function
```

```
End If
For FormatID = 0 To GraphicsFormats.Count - 1
   Set GraphicsFormat = GraphicsFormats.Item(FormatID)
   Dim Ext As String
   Ext = GraphicsFormat.Format
   theArtifact.RenderToFile GraphicsFormat, "File." + Ext
   Results = theArtifact.RenderToFileEx(GraphicsFormat,
"File(Path='FileEx." + Ext + "')|Format(Width=1 Height=1
UnitsPerInch=1)")
Next FormatID
```

Artifact.SetPropertyValue Method

Sets a specific property value by property type. Algorithms that set properties on large numbers of artifacts should use this mechanism instead of using the Property collection.

SetPropertyValue (IArtifactPropertyType Type, VARIANT value)

Return Value

None

Arguments

Type

An artifact property type object (IArtifactPropertyType interface). The IArtifactPropertyType object is an instance of a property type belonging to this artifact's Type member. This value may be obtained given the name of the property using Type.PropertyTypes.Item (name).

value

A variant that specifies the value to be set. Must be consistent with the data type supported by the associated property type.

Example

```
Dim theArtifact As Artifact
Dim theArtifactPropertyType As ArtifactPropertyType
Set theArtifactPropertyType = theArtifact.Type.PropertyTypes.Item(3)
If theArtifact.Properties.IsModifiable = True Then
   theArtifact.SetPropertyValue theArtifactPropertyType, "Just for
testing"
```

```
End If
```

Artifact.SetPropertyValue_ID Method

SetPropertyValue_ID is intended for custom purposes. See SetPropertyValue.

Sets a specific property value identified by the property type id. Works just like SetPropertyValue. Uses the ClassID of the ArtifactType and PropertyID of the property belonging to that type to specify the property.

SetPropertyValue_ID (LONG ClassID, LONG PropertyID, VARIANT value)

Return Value

None

Arguments

ClassID

A unique id for an ArtifactType.

PropertyID

A unique id for a Property type (IArtifactPropertyType interface). This value may be obtained given the name of the property using Type.PropertyTypes.Item (name).PropertyID

value

A variant that specifies the value to be set. Must be consistent with the data type supported by the associated property type.

Example

```
Dim theArtifactType As ArtifactType
Dim thePropertyID As Long
Dim theClassID As Long
Dim theArtifact As Artifact
Dim theArtifactPropertyType As ArtifactPropertyType
‘Get the ClassID
theClassID = theArtifactType.ClassID
‘Get the PropertyID
thePropertyID = theArtifactPropertyType6.PropertyID

Set theArtifactPropertyType = theArtifact.Type.PropertyTypes.Item(3)
```

```
If theArtifact.Properties.IsModifiable = True Then
theArtifact.SetPropertyValue_ID theClassID, thePropertyID, "Just for
Testing"
End If
```

IArtifactArgument Interface

This is the default interface for the artifact arguments. You use artifact arguments to construct artifact locators. The locators locate artifact types or actual instances of artifacts.

ArtifactArgument Properties

Table 8 lists the ArtifactArgument properties.

Table 8 IArtifactArgument Properties

Method	Description
ArgumentName	Returns the name of the argument.
ArtifactTypeName	Returns the name of the artifact type corresponding to the argument.
DataType	Returns the datatype for this property as a VARTYPE.
DefaultValue	Returns the default value for the argument.
Required	Returns TRUE if the argument is required for constructing the artifact locator. Returns FALSE otherwise.
SemanticDataType	Returns the SemanticDataType for this property as a SemanticDataType enumeration.
Value	Returns the value of the argument.
ValueArtifactType	If method returns Artifact, this call returns the ArtifactType

ArtifactArgument.ArgumentName Property

Returns the name of the argument.

ArtifactArgument

Return Value

Returns an argument name.

Arguments

None

Example

```
Dim theLocator As ArtifactLocator
Dim theArgument As ArtifactArgument
Dim ArgID As Integer
Dim theListItem As ListItem

For ArgID = 0 To theLocator.Arguments.Count - 1
   Set theArgument = theLocator.Arguments.Item(ArgID)
   Set theListItem = ListView1.ListItems.Add(Text:=theArgument.Value)
   theListItem.SubItems(1) = theArgument.ArgumentName
   theListItem.SubItems(2) = theArgument.ArtifactTypeName
Next ArgID
```

ArtifactArgument.ArtifactTypeName Property

Returns the name of the artifact type corresponding to the argument.

ArtifactTypeName

Return Value

Returns an artifact type name.

Arguments

None

Example

```
Dim theArtifactTypeArgument As ArtifactArgument
Dim theArtifactTypeName As String
theArtifactTypeName = theArtifactTypeArgument.ArtifactTypeName
```

ArtifactArgument.DataType Property

Returns the datatype for this property as a VARTYPE.

DataType

Return Value

Returns the data type of the artifact argument.

Arguments

None

Example

```
Dim DataTypeName As String

  DataTypeName = Cstr(theArtifactArgumentType.DataType)
  Select Case thePropertyType.DataType
  Case 8 ' VT_BSTR
    DataTypeName = "String"
  Case 11 ' VT_BOOL
    DataTypeName = "Boolean"
  Case 3 ' VT_I4
    DataTypeName = "Integer"
  Case 4 ' VT_R4
  Case 5 ' VT_R8
    DataTypeName = "Real"
  Case 7 ' VT_DATE
    DataTypeName = "Date"
  Case Else
    DataTypeName = CStr(thePropertyType.DataType)
  End Select
```

ArtifactArgument.DefaultValue Property

Returns the default value for the argument.

DefaultValue

Return Value

A Variant reference to the default value for an artifact argument.

Arguments

None

Example

```
Dim DefaultValText As String
Dim theArgument7 As ArtifactArgument
DefaultValText = theArgument7.DefaultValue
```

ArtifactArgument.Required Property

Returns TRUE if the argument is required for constructing an artifact locator or artifact type locator. Returns FALSE if the argument is not required to locate an artifact or artifact type..

Required

Return Value

Returns TRUE if the argument is required for constructing an artifact locator or artifact type locator. Returns FALSE otherwise.

Arguments

None

Example

```
Dim theArgument As ArtifactArgument
'ValueText = JustifyString(theArgument.Value, ValueWidth, True)
   If theArgument.Required = True Then
      RequiredText = JustifyString("Yes", RequiredWidth, True)
   Else
      RequiredText = JustifyString("No", RequiredWidth, True)
   End If
```

ArtifactArgument.SemanticDataType Property

Returns the SemanticDataType for this property as a SemanticDataType enumeration.

SemanticDataType

Return Value

Returns the semantic data type (eSemanticDataType) of an artifact argument.

The semantic data types are: `rsDataObject, rsDescription, rsDirectory, rsFileMoniker, rsFileOrDirectory, rsFilePath, rsName, rsNone, rsPassword, rsURL, rsUserName`. The default is `frwNone`.

Arguments

None

Example

```
Dim SemDataTypeText As String
Dim theArgument As ArtifactArgument
SemDataTypeText = GetSemanticTypeName(theArgument.SemanticDataType)
...
Function GetSemanticTypeName(SemanticDataType As eSemanticDataType) As
String
  Select Case SemanticDataType
  Case rsDirectory
    GetSemanticTypeName = "Directory"
  Case rsFileOrDirectory
    GetSemanticTypeName = "FileOrDirectory"
  Case rsFilePath
    GetSemanticTypeName = "FilePath"
  Case rsName
    GetSemanticTypeName = "Name"
  Case rsUserName
    GetSemanticTypeName = "UserName"
  Case rsPassword
    GetSemanticTypeName = "Password"
  'Case None
  End Select
End Function
```

ArtifactArgument.Value Property

Gets or sets the value of an artifact argument. Set the value by passing in a variant value. Getting the argument value returns a variant reference to the value.

Value (VARIANT Value)

Value

Return Value

Returns a Variant argument value for the ArtifactArgument.

Argument

Value

A Variant value for setting an artifact argument value.

Example

```
Dim theGetVal7 As String
Dim theSetVal7 As Variant
theGetVal7 = theArgument7.Value
theSetVal7 = theArtifactLocator7.Arguments.Item(2).Value()
```

ArtifactArgument.ValueArtifactType Property

If IsValueArtifact returns TRUE, thenValueArtifactType returns the ArtifactType for the Artifact. Otherwise, it returns Zero.

ValueArtifactType

Return Value

A reference to an ArtifactType if the ArtifactArgument is an Artifact.

Arguments

None

Example

```
Dim theArtifactArgument As ArtifactArgument
Dim theArtifactType As ArtifactType
If theArtifactArgument.IsValueArtifact
   Then
   theArtifactType = theArtifactArgument.ValueArtifactType
Else
   ‘do other business logic
```

ArtifactArgument Methods

ArtifactArgument.IsValueArtifact Method

Returns TRUE if the ArtifactArgument is an Artifact. Returns FALSE otherwise.

IsValueArtifact

Return Value

Returns TRUE if the ArtifactArgument is an Artifact. Returns FALSE otherwise.

Arguments

None

Example

```
Dim theArtifactArgument As ArtifactArgument
If theArtifactArgument.IsValueArtifact
   Then
   ‘do business logic on the Artifact
Else
   ‘do other business logic
```

IArtifactArgumentCollection Interface

This is the default interface for the collection of arguments for an artifact.

You create an artifact argument collection object, in VB script, as follows:

```
Dim theSession
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating an Artifact Argument Collection
Dim theArtifactArgumentCollection
Set theArtifactArgumentCollection =
WScript.CreateObject("RDSICore.ArtifactArgumentCollection.1")
```

ArtifactArgumentCollection Properties

Table 9 lists the ArtifactArgumentCollection properties.

Table 9 ArtifactArgumentCollection Properties

Method	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

ArtifactArgumentCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim theArgumentCollectionCount8 As Integer
```

```
Dim theArtifactArgColl81 As ArtifactArgumentCollection
theArgumentCollectionCount8 = theArtifactArgColl81.Count
```

ArtifactArgumentCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns an ArtifactArgument object (IArtifactArgument interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theArtifactLocator8 As ArtifactLocator
Dim theArgument8 As ArtifactArgument
Dim theArtifactArgColl8 As ArtifactArgumentCollection
For ArgID8 = 0 To theArtifactLocator8.Arguments.Count - 1
  Set theArgument8 = theArtifactLocator8.Arguments.Item(ArgID8)
Next ArgID8
Set theArtifactArgColl8 = theArtifactLocator8.Arguments
theArtifactArgColl8.Add theArgument8
```

ArtifactArgumentCollection Methods

Table 10 lists the ArtifactArgumentCollection methods.

Table 10 IArtifactArgumentCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.

Method	Description
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

ArtifactArgumentCollection.Add Method

Appends the specified object to the end of the collection.

Add (IArtifactArgument ArtifactArgument)

Return Value

None

Argument

ArtifactArgument

An ArtifactArgumentCollection object (IArtifactArgumentCollection interface).

Example

```
Dim theArtifactArgColl8 As ArtifactArgumentCollection
Dim theArtifactLocator8 As ArtifactLocator
Dim theArgument8 As ArtifactArgument
For ArgID8 = 0 To theArtifactLocator8.Arguments.Count - 1
  Set theArgument8 = theArtifactLocator8.Arguments.Item(ArgID8)
Next ArgID8
Set theArtifactArgColl8 = theArtifactLocator8.Arguments
theArtifactArgColl8.Add theArgument8
```

ArtifactArgumentCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IArtifactArgumentCollection Collection)

Return Value

None

Argument

Collection

An ArtifactArgumentCollection object (IArtifactArgumentCollection interface).

Example

```
Dim theArtifactArgColl8 As ArtifactArgumentCollection
Dim theArtifactLocator8 As ArtifactLocator
Dim theArtifactArgColl81 As ArtifactArgumentCollection
Set theArtifactArgColl81 = theArtifactLocator8.Arguments
theArtifactArgColl81.AddCollection theArtifactArgColl8
```

ArtifactArgumentCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user-created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theArtifactArgColl81 As ArtifactArgumentCollection
Dim theModifyParam8 As Boolean
theModifyParam8 = theArtifactArgColl81.IsModifiable
```

ArtifactArgumentCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim theArtifactArgColl81 As ArtifactArgumentCollection
theArtifactArgColl81.Remove (0)
theArtifactArgColl81.Remove (1)
```

IArtifactCollection Interface

This is the default interface for the collection of artifacts. There is an artifact collection for each RSE adapter.

You create an artifact collection object, in VB script, as follows:

```
Dim theSession
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating an Artifact Collection
Dim theArtifactCollection
Set theArtifactCollection =
WScript.CreateObject("RDSICore.ArtifactCollection.1")
```

ArtifactCollection Properties

Table 11 lists the ArtifactCollection properties.

Table 11 ArtifactCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

ArtifactCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim theArtCollCount9 As Integer
Dim theArtifactCollection91 As ArtifactCollection
theArtCollCount9 = theArtifactCollection91.Count
```

VB Script Example

```
Dim theSession, theArtifact
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating an Artifact Collection
Dim theArtifactCollection, theCount
Set theArtifactCollection =
WScript.CreateObject("RDSICore.ArtifactCollection.1")
theCount = theArtifactCollection.Count
```

ArtifactCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns an Artifact object (IArtifact interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theArtifact92 As Variant
Dim theArgId9 As Integer
Dim theArtifactCollection9 As ArtifactCollection
For ArgID = 0 To theArtifactCollection9.Count - 1
  Set theArtifact92 = theArtifactCollection9.Item(ArgID)
Next ArgID
```

ArtifactCollection Methods

Table 12 lists the ArtifactCollection methods.

Table 12 IArtifactCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
GetContainedArtifactType	Returns the ArtifactType of the Artifacts contained in the ArtifactCollection.
GetIterator	Returns an iterator for going through the list of items in a collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

ArtifactCollection.Add Method

Appends the specified object to the end of the collection.

Add (IArtifact Artifact)

Return Value

None

Argument

Artifact

An Artifact object (IArtifact interface).

Example

```
Dim theArtifact9 As Artifact
Dim theArtifactCollection9 As ArtifactCollection
Set theArtifactCollection9 = New ArtifactCollection
theArtifactCollection9.Add theArtifact9
```

VB Script Example

```
Dim theSession, theArtifact
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating an Artifact Collection
Dim theArtifactCollection
Set theArtifactCollection =
WScript.CreateObject("RDSICore.ArtifactCollection.1")
```

theArtifactCollection.Add theArtifact

ArtifactCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IArtifactCollection Collection)

Return Value

None

Argument

Collection

Returns an ArtifactCollection object (IArtifactCollection interface).

Example

```
Dim theArtifactCollection9 As ArtifactCollection
Dim theArtifactCollection91 As ArtifactCollection
Set theArtifactCollection91 = New ArtifactCollection
theArtifactCollection91.AddCollection theArtifactCollection9
```

ArtifactCollection.GetContainedArtifactType Method

Returns the ArtifactType of the Artifacts contained in the ArtifactCollection.

In order to support the product-specific generated code, the ArtifactCollection needs to know the ArtifactType of Artifacts that it contains. This must be possible even if a particular collection instance contains no Artifacts. An ArtifactType of NULL means that it could contain Artifacts of more than one ArtifactType. The generated code makes use of this information in order to instantiate the appropriate product-specific collection that inherits from ArtifactCollection.

GetContainedArtifactType

Return Value

Returns a reference to a contained ArtifactType object (IArtifactType interface).

Arguments

None

Example

ArtifactCollection.GetIterator Method

Returns an iterator for going through the list of items in a collection.

GetIterator

Return Value

Returns an ArtifactIterator object (IArtifactIterator interface).

Arguments

None

Example

```
Dim theIterator9 As Variant
Dim theArtifact91 As Artifact
Dim theArtifactCollection9 As ArtifactCollection
Set theIterator9 = theArtifactCollection9.GetIterator
While (theIterator9.HasNext())
  Set theArtifact91 = theIterator9.Next()
Wend
```

ArtifactCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theRetModVal9 As Boolean
Dim theArtifactCollection9 As ArtifactCollection
theRetModVal9 = theArtifactCollection9.IsModifiable
```

ArtifactCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim theArtifactCollection91 As ArtifactCollection
theArtifactCollection91.Remove (0)
```

IArtifactFilter Interface

An instance of IArtifactFilter is a 'filter object'. This composite filter object contains subfilters (which are ArtifactFilters themselves) and atomic expressions. Component filters may be ANDed or ORed together.

ArtifactFilter Properties

Table 13 lists the ArtifactFilter properties.

Table 13 ArtifactFilter Properties

Method	Description
ArtifactType	Returns an artifact type for a filter.
Comparisons	Returns the collection of filter comparison objects.
Filters	Returns the IArtifactFilterCollection.
LogicalOp	Returns the logical operator for a filter. The default is the AND operator.

ArtifactFilter.ArtifactType Property

Returns an artifact type for a filter. Composite filter object, contains subfilters (which are ArtifactFilters themselves)and atomic expressions. Component filters may be ANDed or ORed together.

ArtifactType

Return Value

Returns an ArtifactType object (IArtifactType interface).

Arguments

None

Example

```
Dim theFilter As ArtifactFilter
Dim theFilterArtifactType As ArtifactType
```

```
theFilterArtifactType = theFilter.ArtifactType
```

ArtifactFilter.Comparisons Property

Returns the collection of filter comparison objects (IFilterComparsonCollection).

Comparisons

Return Value

Returns a FilterComparisonCollection object (IFilterComparisonCollection interface).

Arguments

None

Example

```
Dim theFilter As ArtifactFilter
Dim theComparisons As FilterComparisonCollection
Set theComparisons = theFilter.Comparisons
```

ArtifactFilter.Filters Property

Returns a collection of ArtifactFilter objects (IArtifactFilterCollection interface).

Filters

Return Value

Returns an ArtifactFilterCollection object (IArtifactFilterCollection interface).

Arguments

None

Example

```
Dim theFilter As ArtifactFilter
Dim theFilters As ArtifactFilterCollection
Set theFilters = theFilter.Filters
```

ArtifactFilter.LogicalOp Property

Gets or sets a logical operator for an artifact filter. Returns the logical operator for a filter. The default is the AND operator.

Set a logical operator for a filter by providing a new logical operator value.

LogicalOp

LogicalOp (LogicalOperator newVal)

Return Value

Returns a logical operator.

Argument

newVal

A value of type LogicalOperator (for example, LOGICAL_OP_AND)

Example

```
Dim theFilter As ArtifactFilter

'The following gets a Logical Operator
Dim theNewFilter As ArtifactFilter
Dim theRelationship As RelationshipType
Set theNewFilter = theRelationship.CreateArtifactFilter
theNewFilter.LogicalOp = theFilter.LogicalOp

'The following sets a logical operator
Dim theLogicalOp As LogicalOperator
theLogicalOp = theFilter.LogicalOp(LOGICAL_OP_OR)
```

ArtifactFilter Methods

Table 14 lists the IArtifactFilter methods.

Table 14 IArtifactFilter Methods

Method	Description
AddComparison	Adds a new simple comparison filter to the filter.
AddFilter	Adds a new subfilter to the filter.
ArtifactApplies	Evaluates the filter with regard to the provided artifact.

Method	Description
DeleteComparison	Deletes a comparison out of the list of comparisons.
DeleteFilter	Deletes a filter out of the list of filters
GetFilterString	Returns a query string for a filter
SetFilterString	Sets query string for building all the hierarchy of comparisons and filters

ArtifactFilter.AddComparison Method

Add a new simple comparison filter (a FilterComparison object) to the filter.

See the *Developing Applications with Rational Suite Extensibility* for the complete list of comparison operators.

AddComparison(BSTR PropertyName, ComparisonOperator ComparisonOp, VARIANT Val)

Return Value

Returns a FilterComparison object (IFilterComparison interface).

Arguments

PropertyName

A BSTR property name. This must be a valid name of one of the properties of the artifact being evaluated.

ComparisonOp

A ComparisonOperator (for example, COMP_OP_NOT_BETWEEN)

Val

A Variant filter comparison value. This is the value of the property specified in the PropertyPath argument. For example, if the Property name is Path, then the value could be "C:\pathname\filename.doc."

Example

```
   Dim theOldFilter As ArtifactFilter
Dim theNewFilter As ArtifactFilter
Dim theComparisons As FilterComparisonCollection
```

```
Dim theComparison As FilterComparison
Dim theTargetComparison As FilterComparison

Set theComparisons = theOldFilter.Comparisons
For Index = 0 To theComparisons.Count - 1
  Set theComparison = theComparisons.Item(Index)
  Set theTargetComparison = theNewFilter.AddComparison(
                                        theComparison.PropertyName,
                                    theComparison.ComparisonOp,
                                    theComparison.Value)
Next Index
```

ArtifactFilter.AddFilter Method

Adds a new subfilter to the filter.

AddFilter (LogicalOperator LogicalOp)

Return Value

Returns an ArtifactFilter object (IArtifactFilter interface).

Argument

LogicalOp

A Logical Operator (AND or OR).

Example

```
Dim theOldFilter As ArtifactFilter
Dim theNewFilter As ArtifactFilter
For Index = 0 To theOldFilter.Filters.Count - 1
   Set SourceFilter = theOldFilter.Filters.Item(Index)
   Set TargetFilter = theNewFilter.AddFilter(SourceFilter.LogicalOp)
   Duplicate_Filter SourceFilter, TargetFilter
Next Index
```

ArtifactFilter.ArtifactApplies Method

Evaluate the filter with regard to the provided artifact.

This method determines if the criteria set in the filter object (the interface this method is defined in) applies to the given artifact. The artifact to be evaluated is passed in as a parameter to the method. The filter is then evaluated using the provided artifact as the data.

ArtifactApplies (IArtifact Artifact)

Return Value

Returns TRUE if the filter applies to the Artifact. Empty AND filter evaluates to TRUE. Empty OR filter evaluates to FALSE.

Argument

Artifact

An Artifact object (IArtifact interface).

Example

```
Dim theFilter As ArtifactFilter
Dim theArtifact As Artifact
Dim Result As Boolean
Result = theFilter.ArtifactApplies(theArtifact)
```

ArtifactFilter.DeleteComparisons Method

Deletes a comparison out of the list of comparisons.

DeleteComparison (IFilterComparison Comparison)

Return Value

None

Argument

Comparison

A FilterComparison object (IFilterComparison interface).

Example

```
Dim theFilter As ArtifactFilter
Dim theTargetComparison As FilterComparison
theFilter.DeleteComparison(theTargetComparison)
```

ArtifactFilter.DeleteFilter Method

Deletes a filter out of the list of filters.

DeleteFilter (IArtifactFilter Filter)

Return Value

None

Argument

Filter

An ArtifactFilter object (IArtifactFilter interface).

Example

```
Dim theFilter As ArtifactFilter
Dim theTargetArtifactFilter As ArtifactFilter
theFilter.DeleteFilter(theTargetArtifactFilter)
```

ArtifactFilter.GetFilterString Method

Returns a query string for a filter.

GetFilterString

Return Value

Returns a query string.

Arguments

None

Example

```
Dim theFilter As ArtifactFilter
Dim theFilterString As String
theFilterString = theFilter.GetFilterString
PrintComment 0, "Getting Filter String:"
PrintComment 1, theFilterString
```

ArtifactFilter.SetFilterString Method

Sets a query string for building the hierarchy of comparisons and filters.

SetFilterString (BSTR newVal)

Return Value

None

Argument

newVal

A query string

Example

```
Dim theFilter As ArtifactFilter
Dim FilterString As String
theRelationship As RelationshipType
Set theFilter = theRelationship.CreateArtifactFilter
theFilter.SetFilterString (FilterString)
```

IArtifactFilterCollection Interface

A collection of filters. No modification to the the collection is needed.

You create an artifact filter collection object, in VB script, as follows:

```
Dim theSession
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating an Artifact Filter Collection
Dim theArtifactFilterCollection
Set theArtifactFilterCollection =
WScript.CreateObject("RDSICore.ArtifactFilterCollection.1")
```

ArtifactFilterCollection Properties

Table 15 lists the ArtifactFilterCollection properties.

Table 15 IArtifactFilterCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

ArtifactFilterCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

A reference to the count or number of items in the collection.

Arguments

None

Example

```
Dim theArtifactFilterCollection As ArtifactFilterCollection
```

```
Dim theArtFilterCollCount9 As Integer
theArtFilterCollCount9 = theArtifactFilterCollection.Count
```

ArtifactFilterCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

An ArtifactFilter object (IArtifactFilter interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theFilters As ArtifactFilterCollection
Dim theFilter As ArtifactFilter
Set theFilters = theFilter.Filters
FCount = theFilters.Count

If FCount > 0 Then
   For Index = 0 To FCount - 1
      Set theFilter = theFilters.Item(Index)
   Next Index
End If
```

ArtifactFilterCollection Methods

Table 16 lists the IArtifactFilterCollection methods.

Table 16 IArtifactFilterCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.

Method	Description
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

ArtifactFilterCollection.Add Method

Appends the specified object to the end of the collection.

Add (IArtifactFilter ArtifactFilter)

Return Value

None

Arguments

ArtifactFilter

An ArtifactFilter object (IArtifactFilter interface).

Example

```
Dim theArtifactFilter As ArtifactFilter
Dim theArtifactFilterCollection As ArtifactFilterCollection
Set theArtifactFilterCollection = New ArtifactFilterCollection
theArtifactFilterCollection.Add theArtifactFilter
```

ArtifactFilterCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IArtifactFilterCollection Collection)

Return Value

None

Argument

Collection

An ArtifactFilterCollection object.

Example

```
Dim theArtifactFilterCollection As ArtifactFilterCollection
Dim theArtifactFilterCollection2 As ArtifactFilterCollection
Set theArtifactFilterCollection = New ArtifactFilterCollection
theArtifactFilterCollection.AddCollection theArtifactFilterCollection2
```

ArtifactFilterCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theArtifactFilterCollection As ArtifactFilterCollection
Dim theRetModVal As Boolean
theRetModVal = theArtifactFilterCollection.IsModifiable
```

ArtifactFilterCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim theFilters As ArtifactFilterCollection
theFilters.Remove (0)
```

IArtifactGraphicsFormatType Interface

Describes the details of the format used by the artifact to render itself.

ArtifactGraphicsFormatType Properties

Table 17 lists the IArtifactGraphicsFormatType properties.

Table 17 IArtifactGraphicsFormatType Properties

Method	Description
Description	Returns the description of the artifact graphics format type.
Format	Returns the format name of the artifact graphics format type.
Name	Returns the name of the artifact graphics format type.

ArtifactGraphicsFormatType.Description Property

Returns the description of the artifact graphics format type.

Description

Return Value

Returns the description of the graphics format type of the artifact.

Arguments

None

Example

```
Dim theGraphicsFormats12 As ArtifactGraphicsFormatTypeCollection
Dim theGraphicsFormat12 As ArtifactGraphicsFormatType
Dim theArtifact12 As Artifact
Dim theGFormatID12 As Integer
Dim theDesc12 As Variant

Set theGraphicsFormats12 = theArtifact12.Type.GraphicsFormats
For theGFormatID12 = 0 To theGraphicsFormats12.Count - 1
```

```
  Set theGraphicsFormat12 = theGraphicsFormats12.Item(theGFormatID12)
  theDesc12 = theGraphicsFormat12.Description
Next theGFormatID12
```

ArtifactGraphicsFormatType.Format Property

Returns the format name of the artifact graphics format type.

Format

Return Value

Returns the format name of the artifact graphics format type.

Arguments

None

Example

```
Dim theGraphicsFormats12 As ArtifactGraphicsFormatTypeCollection
Dim theGraphicsFormat12 As ArtifactGraphicsFormatType
Dim theArtifact12 As Artifact
Dim theGFormatID12 As Integer
Dim theFormat12 As Variant

Set theGraphicsFormats12 = theArtifact12.Type.GraphicsFormats
For theGFormatID12 = 0 To theGraphicsFormats12.Count - 1
   Set theGraphicsFormat12 = theGraphicsFormats12.Item(theGFormatID12)
   theFormat12 = theGraphicsFormat12.Format
Next theGFormatID12
```

ArtifactGraphicsFormatType.Name Property

Returns the name of the artifact graphics format type.

Name

Return Value

Returns the name of the artifact graphics format type.

Arguments

None

Example

```
Dim theGraphicsFormats12 As ArtifactGraphicsFormatTypeCollection
Dim theGraphicsFormat12 As ArtifactGraphicsFormatType
Dim theArtifact12 As Artifact
Dim theGFormatID12 As Integer
Dim theName12 As Variant

Set theGraphicsFormats12 = theArtifact12.Type.GraphicsFormats
For theGFormatID12 = 0 To theGraphicsFormats12.Count - 1
  Set theGraphicsFormat12 = theGraphicsFormats12.Item(theGFormatID12)
  theName12 = theGraphicsFormat12.Name
Next theGFormatID12
```

IArtifactGraphicsFormatTypeCollection Interface

This is the default interface for the collection of graphics format types for an artifact.This interface encapsulates information about the list of artifact graphics format types.

You create an artifact graphics format type collection object, in VB script, as follows:

```
Dim theSession
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating an Artifact Graphics Format Type Collection
Dim theArtifactGraphicsFormatTypeCollection
Set theArtifactGraphicsFormatTypeCollection =
WScript.CreateObject("RDSICore.ArtifactGraphicsFormatTypeCollection.1"
)
```

ArtifactGraphicsFormatTypeCollection Properties

Table 18 lists the ArtifactGraphicsFormatTypeCollection properties.

Table 18 ArtifactGraphicsFormatTypeCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

ArtifactGraphicsFormatTypeCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim theGraphicsFormats13 As ArtifactGraphicsFormatTypeCollection
Dim theCount13 As Integer
theCount13 = theGraphicsFormats13.Count
```

ArtifactGraphicsFormatTypeCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns the ArtifactGraphicsFormatType object value at the specified index (ArtifactGraphicsFormatType interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theGraphicsFormats13 As ArtifactGraphicsFormatTypeCollection
Dim theGraphicsFormat As ArtifactGraphicsFormatType
Set theGraphicsFormat = theGraphicsFormats13.Item(0)
```

ArtifactGraphicsFormatTypeCollection Methods

Table 19 lists the ArtifactGraphicsFormatTypeCollection methods.

Table 19 IArtifactGraphicsFormatTypeCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

ArtifactGraphicsFormatTypeCollection.Add Method

Appends the specified object to the end of the collection.

Add (IArtifactGraphicsFormatType GraphicsFormatType)

Return Value

None

Argument

GraphicsFormatType

An ArtifactGraphicsFormatType object (IArtifactGraphicsFormatType interface).

Example

```
Dim theGraphicsFormats13 As ArtifactGraphicsFormatTypeCollection
Dim theGraphicsFormat13 As ArtifactGraphicsFormatType
theGraphicsFormats13.Add theGraphicsFormat13
```

ArtifactGraphicsFormatTypeCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IArtifactGraphicsFormatTypeCollection Collection)

Return Value

None

Argument

Collection

An ArtifactGraphicsFormatTypeCollection object (IArtifactGraphicsFormatTypeCollection interface).

Example

```
Dim theGraphicsFormats131 As ArtifactGraphicsFormatTypeCollection
Dim theGraphicsFormats13 As ArtifactGraphicsFormatTypeCollection
theGraphicsFormats131.AddCollection theGraphicsFormats13
```

ArtifactGraphicsFormatTypeCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theGraphicsFormats13 As ArtifactGraphicsFormatTypeCollection
Dim theRetval As Boolean
theRetval = theGraphicsFormats13.IsModifiable
```

ArtifactGraphicsFormatTypeCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim theGraphicsFormats13 As ArtifactGraphicsFormatTypeCollection
theGraphicsFormats13.Remove(0)
```

IArtifactGUI Interface

IArtifactGUI is supported by Artifact objects that are capable of displaying themselves in a GUI.

ArtifactGUI Methods

Table 20 lists the IArtifactGUI methods.

Table 20 IArtifactGUI Methods

Method	Description
CanShow	Returns FALSE if the Artifact cannot display itself in a GUI, TRUE otherwise.
Show	Instructs the Artifact to make its GUI visible.

ArtifactGUI.CanShow Method

Returns TRUE if the artifact can display itself in a GUI, FALSE otherwise. Call this method before calling ArtifactGUI.Show.

CanShow (BOOL CanShow)

Return Value

Returns TRUE if the artifact can display itself in a GUI, FALSE otherwise.

Arguments

None

Example

```
Dim theArtifact As Artifact
' theArtifact.GUI returns an artifact GUI object
If theArtifact.GUI.CanShow() Then
   PrintResults 0, "Artifact can show. Calling theArtifact.GUI.Show..."
Else
   PrintResults 0, "Artifact can't show itself
(theArtifact.GUI.CanShow() returned false)."
```

```
End If
```

ArtifactGUI.Show Method

Instructs the Artifact to make its GUI visible.

Show

Return Value

None

Arguments

None

Example

```
Dim theArtifact As Artifact
If theArtifact.GUI.CanShow() Then
   theArtifact.GUI.Show
End If
```

IArtifactIterator Interface

Iterates through a given artifact collection. No modification to a collection is needed to iterate through the collection. Each iterator begins iteration from the beginning of the collection.

Rational recommends that clients convert code as needed to use the iterators, especially where performance is critical and for large collections.

The IArtifactCollection GetIterator method returns a new instance of an IArtifactIterator.

ArtifactIterator Methods

Table 21 lists the IArtifactIterator methods.

Table 21 IArtifactIterator Methods

Method	Description
HasNext	Returns TRUE if the collection has another item to iterate.
Next	Returns the next item in the collection.

ArtifactIterator.HasNext Method

Returns TRUE if the collection has another item to iterate.

HasNext

Return Value

Returns TRUE if the collection has another item to iterate, FALSE otherwise.

Arguments

None

Example

```
Sub IterateArtifacts (theArtifacts As ArtifactCollection)
Dim Iterator As ArtifactIterator
Set Iterator = theArtifacts.GetIterator
```

```
While Iterator.HasNext ()
   Set theArtifact = Iterator.Next ()
WEnd
End Sub
```

ArtifactIterator.Next Method

Returns the next item in the collection.

Next (IArtifact Artifact)

Return Value

Returns an Artifact object (IArtifact interface).

Arguments

None

Example

```
Sub IterateArtifacts (theArtifacts As ArtifactCollection)
Dim Iterator As ArtifactIterator
Set Iterator = theArtifacts.GetIterator
While Iterator.HasNext ()
   Set theArtifact = Iterator.Next ()
WEnd
End Sub
```

IArtifactLocator Interface

The default interface of the artifact locator object.

ArtifactLocator Properties

Table 22 lists the ArtifactLocator properties.

Table 22 IArtifactLocator Properties

Property	Description
Arguments	Returns the collection of arguments that apply to this locator.
AvailableLocatorTypes	Returns the list of locator types that can be used for this artifact.
ContextLocator	Returns a context locator object.
LocatorType	Returns the type of locator that this locator is using.

ArtifactLocator.Arguments Property

Returns the collection of arguments that apply to this locator.

Arguments

Return Value

Returns an ArtifactArgumentCollection object (IArtifactArgumentCollection interface).

Arguments

None

Example

```
Dim theLocator As ArtifactLocator
Dim theArgument As ArtifactArgument
Dim ArgID As Integer
Dim theListItem As ListItem
For ArgID = 0 To theLocator.Arguments.Count - 1
```

```
   Set theArgument = theLocator.Arguments.Item(ArgID)
   Set theListItem = ListView1.ListItems.Add(Text:=theArgument.Value)
   theListItem.SubItems(1) = theArgument.ArgumentName
   theListItem.SubItems(2) = theArgument.ArtifactTypeName
Next ArgID
```

ArtifactLocator.AvailableLocatorTypes Property

Returns the list of locator types that can be used for this artifact.

AvailableLocatorTypes (IArtifactLocatorTypeCollection Types)

Return Value

Returns an ArtifactLocatorTypeCollection object (IArtifactLocatorTypeCollection interface).

Arguments

None

Example

```
Dim theLocator As ArtifactLocator
Dim theAvailableTypes As ArtifactLocatorTypeCollection
Set theAvailableTypes = theLocator.AvailableLocatorTypes
```

ArtifactLocator.ContextLocator Property

Returns a context locator object. The context locator context can then be used to get a relative artifact ID (GetRelativeArtifactID).

ContextLocator

Return Value

Returns a context ArtifactLocator object (IArtifactLocator interface).

Arguments

None

Example

```
Dim theLocator As ArtifactLocator
Dim theContextLocator As ArtifactLocator
```

```
Set theContextLocator = theLocator.ContextLocator
```

ArtifactLocator.LocatorType Property

Gets or Sets the LocatorType property of an ArtifactLocator object. Setting the locator type changes the list of arguments available for this locator.

LocatorType

Returns the type of locator that this locator is using.

LocatorType (IArtifactLocatorType Type)

Sets the artifact locator type. Setting the locator type changes the list of arguments available for this locator.

Return Value

Returns the type of locator the locator object.

Argument

Type

An artifact locator type (IArtifactLocatorType interface).

Example

```
Dim theLocatorType As ArtifactLocatorType
Dim theLocator As ArtifactLocator
theLocatorType = theLocator.LocatorType
If theLocatorType = SupportsLocatorType(rsLocator_Display_Name) Then
...
If theLocator = IsDefault(rsLocator_Display_Name) Then
...
```

ArtifactLocator Methods

Table 23 lists the IArtifactLocator methods.

Table 23 IArtifactLocator Methods

Method	Description
CreateRelativeLocator	Returns a relative locator to an artifact, given a relative artifact id.

Method	Description
GetArtifactID	Returns an artifact id for this locator.
GetRelativeArtifactID	Returns a relative artifact id for this locator.
IsResolved	Returns TRUE if the Locator has successfully located its artifact. Returns FALSE otherwise.
LocateArtifact	Returns the Artifact object identified by the specified artifact locator.
LocateInternalObject	Returns the IUnknown for the internal object identified by the specified ArtifactID string.

ArtifactLocator.CreateRelativeLocator Method

Returns a relative locator to an artifact, given a relative artifact ID.

CreateRelativeLocator (BSTR RelativeArtifactID)

Return Value

Returns an ArtifactLocator object (IArtifactLocator interface).

Argument

RelativeArtifactID

An artifact ID that is relative to a context artifact locator.

Example

```
Dim RelArtifactID As String
Dim theLocator As ArtifactLocator
Dim theContextLocator As ArtifactLocator
theContextLocator = theLocator.ContextLocator
RelArtifactID = theLocator.GetRelativeArtifactID(theContextLocator,
rsLocator_Display_Name)
Set theLocator =
theContextLocator.CreateRelativeLocator(RelArtifactID)
```

ArtifactLocator.GetArtifactID Method

Returns an artifact id for this locator. The artifact id contains the string representation of the arguments and property values for this locator.

GetArtifactID (eLocatorType Type)

Return Value

Returns the ArtifactID as a string.

Argument

Type

An optional argument specifying the artifact locator type (for example, rsLocator_Display_Name).

Example

```
Dim theLocator As ArtifactLocator
Dim LocatorString As String
LocatorString = theLocator.GetArtifactID(rsLocator_Display_Name)
```

ArtifactLocator.GetRelativeArtifactID Method

Returns a relative artifact id for this locator. The relative id contains the arguments and property values necessary to locate the relative artifact relative to the context artifact.

The context artifact is the artifact that is used as the context for the relative ID. This artifact is needed to resolve the relative ID.

GetRelativeArtifactID (IArtifactLocator ContextLocator, eLocatorType Type)

Return Value

Returns an artifact ID as a string value.

Arguments

ContextLocator

The locator to the context artifact.

Type

The type of locator to return. Either Display Name or UniqueID. This argument is optional.

Example

```
Dim RelArtifactID As String
```

```
Dim theLocator As ArtifactLocator
Dim theContextLocator As ArtifactLocator
RelArtifactID = theLocator.GetRelativeArtifactID(theContextLocator,
rsLocator_Display_Name)
```

ArtifactLocator.IsResolved Method

Returns TRUE if the locator successfully locates its artifact on the LocateArtifact method. Returns FALSE otherwise.

IsResolved

Arguments

None

Example

```
Dim theArtifact As Artifact
Dim theLocator As ArtifactLocator
Set theArtifact = theLocator.LocateArtifact
If theLocator.IsResolved Then
PrintComment 0, "LocateArtifact found an artifact using
theLocator.GetArtifactID(rsLocator_Display_Name)
End If
```

ArtifactLocator.LocateArtifact Method

Returns the IArtifact for the Artifact object identified by the specified ArtifactID string.

LocateArtifact

Return Value

Returns an Artifact object (IArtifact interface).

Arguments

None

Example

```
Dim theArtifact As Artifact
Dim theLocator As ArtifactLocator
```

```
Set theArtifact = theLocator.LocateArtifact
```

ArtifactLocator.LocateInternalObject Method

Returns the IUnknown for the internal object identified by the specified ArtifactID string. This allows the client to use RSE Locators to resolve directly to integrated-product objects without creating an instance of an artifact.

LocateInternalObject

Return Value

Returns the IUnknown of the associated internal object.

Arguments

None

Example

```
Dim theArtifactLocator15 As ArtifactLocator
Dim theLocatedObject As Variant
Set theLocatedObject = theArtifactLocator15.LocateInternalObject
```

IArtifactLocatorCollection Interface

This is the default interface for the collection of locators for an artifact.This interface encapsulates information about the list of artifact locators.

You create an artifact locator collection object, in VB script, as follows:

```
Dim theSession
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating an Artifact Locator Collection
Dim theArtifactLocatorCollection
Set theArtifactLocatorCollection =
WScript.CreateObject("RDSICore.ArtifactLocatorCollection.1")
```

ArtifactLocatorCollection Properties

Table 24 lists the ArtifactLocatorCollection properties.

Table 24 ArtifactLocatorCollection Properties

Method	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

ArtifactLocatorCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim theArtifactLocatorCollection As ArtifactLocatorCollection
Dim theCount As Integer
theCount = theArtifactLocatorCollection.Count
```

ArtifactLocatorCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns an ArtifactLocator object (IArtifactLocator interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theArtifactLocatorCollection As ArtifactLocatorCollection
Dim theArtifactLocator As ArtifactLocator
Set theArtifactLocator = theArtifactLocatorCollection.Item(0)
```

ArtifactLocatorCollection Methods

Table 25 lists the ArtifactLocatorCollection methods.

Table 25 IArtifactLocatorCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

ArtifactLocatorCollection.Add Method

Appends the specified object to the end of the collection.

Add (IArtifactLocator ArtifactLocator)

Return Value

None

Argument

ArtifactLocator

An ArtifactLocator object (IArtifactLocator interface).

Example

```
Dim theArtifactLocatorCollection As ArtifactLocatorCollection
Dim theArtifactLocator17 As ArtifactLocator
Set theArtifactLocatorCollection = New ArtifactLocatorCollection
theArtifactLocatorCollection.Add theArtifactLocator17
```

ArtifactLocatorCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IArtifactLocatorCollection Collection)

Return Value

None

Argument

Collection

A reference to an AdapterCollection object (IAdapterCollection interface).

Example

```
Dim theLocatorCollection As ArtifactLocatorCollection
Dim theLocatorCollection2 As ArtifactLocatorCollection
Set theLocatorCollection = New ArtifactLocatorCollection
theLocatorCollection.AddCollection theLocatorCollection2
```

ArtifactLocatorCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theArtifactLocatorCollection As ArtifactLocatorCollection
Dim theRetval As Boolean
theRetval = theArtifactLocatorCollection.IsModifiable
```

ArtifactLocatorCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim theArtifactLocatorCollection As ArtifactLocatorCollection
theArtifactLocatorCollection.Remove(0)
```

IArtifactLocatorType Interface

This is the default interface for the ArtifactLocatorType object. These methods provide information on an artifact locator's type.

ArtifactLocatorType Properties

Table 26 lists the ArtifactLocatorType properties.

Table 26 IArtifactLocatorType Properties

Property	Description
AdapterName	Returns the name of the adapter that owns the locator type.
Arguments	Returns the collection of arguments that apply to this locator type.
LocatorTypeID	Returns the unique ID for the locator type used for indexed lookup.
Name	Returns the descriptive name of the locator.
RelativeArtifactType	Returns the ArtifactType object associated with this locator type.
TypeName	Returns the name of the artifact type associated with this locator type.

ArtifactLocatorType.AdapterName Property

Returns the name of the adapter that owns the locator type.

AdapterName

Return Value

Returns the name of the adapter.

Arguments

None

Example

```
Dim theLocatorType As ArtifactLocatorType
Dim theAdapterName As String
```

```
Set theAdapterName = theLocatorType.AdapterName
```

ArtifactLocatorType.Arguments Property

Returns the collection of arguments that apply to this locator type.

Arguments

Return Value

Returns an ArtifactArgumentCollection object (IArtifactArgumentCollection interface).

Arguments

None

Example

```
Dim theLocatorType As ArtifactLocatorType
Dim theArguments As ArtifactArgumentCollection
Set theArguments = theLocatorType.Arguments
```

ArtifactLocatorType.LocatorTypeID Property

Returns the unique ID for the locator type used for indexed lookup.

LocatorTypeID

Return Value

Returns the unique ID for the locator type used for indexed lookup.

Arguments

None

Example

```
Dim theLocatorTypeID As String
Dim theLocatorType As ArtifactLocatorType
theLocatorTypeID = theLocatorType.LocatorTypeID
```

ArtifactLocatorType.Name Property

Returns the descriptive name of the locator.

Note: This method is currently not supported.

Name

Return Value

Returns the name of the locator.

Arguments

None

Example

```
Dim theArtifactLocatorType As ArtifactLocatorType
Dim theArtifactName As String
theArtifactName = theArtifactLocatorType.Name
```

ArtifactLocatorType.RelativeArtifactType Property

Returns the relative ArtifactType object associated with this locator type.

RelativeArtifactType

Return Value

Returns the relative ArtifactType object associated with this locator type (IArtifactType interface).

Arguments

None

Example

```
Dim theRelativeArtifactType As ArtifactType
Dim theLocatorType As ArtifactLocatorType
If theLocatorType.IsRelative
Then theRelativeArtifactType = theLocatorType.RelativeArtifactType
```

ArtifactLocatorType.TypeName Property

Returns the name of the artifact type associated with this locator type.

TypeName

Return Value

Returns the name of the artifact type.

Arguments

None

Example

```
Dim theArtifactName As String
Dim theLocatorType As ArtifactLocatorType
theArtifactName = theLocatorType.TypeName
```

ArtifactLocatorType Methods

Table 27 lists the IArtifactLocatorType methods.

Table 27 IArtifactLocatorType Methods

Method	Description
IsDefault	Returns TRUE if the locator is the default display name or immutable ID locator.
IsRelative	Returns TRUE if this locator is relative to another artifact type
LocateArtifactType	Returns the ArtifactType object associated with this locator.
SupportsLocatorType	Returns TRUE if the locator is the display name or default immutable ID locator.

ArtifactLocatorType.IsDefault Method

Returns TRUE if the locator is the default display name or immutable ID locator. A custom locator returns FALSE.

IsDefault (eLocatorType Type)

Return Value

Returns TRUE if the given locator is a default display name or immutable ID locator. A custom locator returns FALSE.

Argument

Type

The locator type.

Example

```
Dim theLocatorType As ArtifactLocatorType
If theLocatorType.IsDefault(rsLocator_Display_Name)
   Then
   ...
ElseIf theLocatorType.IsDefault(rsLocator_Immutable_ID)
   Then
   ...
```

ArtifactLocatorType.IsRelative Method

Returns TRUE if this locator is relative to another artifact type.

IsRelative

Return Value

Returns TRUE if this locator is relative to another artifact type. Returns FALSE otherwise.

Arguments

None

Example

```
Dim theLocatorType As ArtifactLocatorType
If theLocatorType.IsRelative
   Then
   '...do business logic
Else
   '...do other business logic
```

ArtifactLocatorType.LocateArtifactType Method

Returns the artifact type object associated with this locator. This call returns a valid object, assuming the locator is referencing valid information.

LocateArtifactType

Return Value

Returns the ArtifactType object (IArtifactType interface) associated with this locator.

Arguments

None

Example

```
Dim theArtifactType As ArtifactType
Dim theLocator As ArtifactLocator
'theLocator.LocatorType returns an instance of a locator type object
Set theArtifactType = theLocator.LocatorType.LocateArtifactType
```

ArtifactLocatorType.SupportsLocatorType Method

Returns TRUE if the locator is the display name or default immutable ID locator.

SupportsLocatorType (eLocatorType Type)

Return Value

Returns TRUE if the locator is the display name or default immutable ID locator, FALSE otherwise.

Argument

Type

The locator type (for example, rsLocator_Display_Name).

Example

```
Dim theLocatorType As ArtifactLocatorType
If theLocatorType.SupportsLocatorType(rsLocator_Display_Name) Then
theLocatorType = rsLocator_Display_Name
```

IArtifactLocatorTypeCollection Interface

This is the default interface for the collection of locator types for an artifact. This interface encapsulates information about the list of artifact locator types.

You create an artifact locator type collection object, in VB script, as follows:

```
Dim theSession
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating an Artifact LocatorType Collection
Dim theArtifactLocatorTypeCollection
Set theArtifactLocatorTypeCollection =
WScript.CreateObject("RDSICore.ArtifactLocatorTypeCollection.1")
```

Table 29 lists the IArtifactLocatorTypeCollection methods.

ArtifactLocatorTypeCollection Properties

Table 28 lists the ArtifactLocatorTypeCollection properties.

Table 28 ArtifactLocatorTypeCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

ArtifactLocatorTypeCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

The count or number of items in the collection.

Arguments

None

Example

```
Dim theLocatorTypeCollection As ArtifactLocatorTypeCollection
Dim theCount As Integer
theCount = theLocatorTypeCollection.Count
```

ArtifactLocatorTypeCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns an ArtifactLocatorType object (IArtifactLocatorType interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theArtifactLocatorTypeCollection As ArtifactLocatorTypeCollection
Dim theArtifactLocatorType As ArtifactLocatorType
Set theArtifactLocatorType = theArtifactLocatorTypeCollection.Item(0)
```

ArtifactLocatorTypeCollection Methods

Table 29 lists the ArtifactLocatorTypeCollection methods.

Table 29 IArtifactLocatorTypeCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

ArtifactLocatorTypeCollection.Add Method

Appends the specified object to the end of the collection.

Add (IArtifactLocatorType ArtifactLocatorType)

Return Value

None

Argument

ArtifactLocatorType

An ArtifactLocatorType object (IArtifactLocatorType interface).

Example

```
Dim theArtifactLocatorType As ArtifactLocatorType
Dim theArtifactLocatorTypeCollection As ArtifactLocatorTypeCollection
Set theArtifactLocatorTypeCollection=New ArtifactLocatorTypeCollection
theArtifactLocatorTypeCollection.Add theArtifactLocatorType
```

ArtifactLocatorTypeCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IArtifactLocatorTypeCollection Collection)

Return Value

None

Argument

Collection

An ArtifactLocatorTypeCollection object (IArtifactLocatorTypeCollection interface).

Example

```
Dim theLocatorTypeCollection As ArtifactLocatorTypeCollection
Dim theLocatorTypeCollection2 As ArtifactLocatorTypeCollection
Set theLocatorTypeCollection = New ArtifactLocatorTypeCollection
theLocatorTypeCollection.AddCollection theLocatorTypeCollection2
```

ArtifactLocatorTypeCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theArtifactLocatorTypeCollection As ArtifactLocatorTypeCollection
Dim theRetval As Boolean
theRetval = theArtifactLocatorTypeCollection.IsModifiable
```

ArtifactLocatorTypeCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim theArtifactLocatorTypeCollection As ArtifactLocatorTypeCollection
theArtifactLocatorTypeCollection.Remove(0)
```

IArtifactPersist Interface

This interface is supported by Artifact objects that define storage-related methods.

ArtifactPersist Methods

Table 30 lists the IArtifactPersist methods.

Table 30 IArtifactPersist Methods

Method	Description
CanRefresh	Returns TRUE if the Artifact supports refresh operations, otherwise it returns FALSE.
CanSave	Returns TRUE if the underlying object supports an explicit Save to storage function. Returns FALSE otherwise.
NeedsRefresh	Returns rsNoRefresh if the object is known to be "up-to-date" relative to the current contents of persistent storage.
NeedsSave	Returns FALSE if the contents of storage are known to be "up-to-date" with the object. Returns TRUE otherwise.
Refresh	Causes data in the current artifact to be refreshed from storage.
Save	Causes data in the current artifact to be written out to persistent storage.

ArtifactPersist.CanRefresh Method

Returns TRUE if the Artifact supports refresh operations, otherwise it returns FALSE.

CanRefresh

Return Value

Returns TRUE if the Artifact supports refresh operations, otherwise it returns FALSE.

Arguments

None

Example

```
Dim theArtifact As Artifact
If theArtifact.Persist.CanRefresh() Then
   Select Case theArtifact.Persist.NeedsRefresh()
   Case rsNoRefresh
    PrintResults 0, "Artifact needs no refresh (rsNoRefresh)."
   Case rsNormalRefresh
    PrintResults 0, "Artifact needs normal refresh (rsNormalRefresh)."
   Case rsQuickRefresh
    PrintResults 0, "Artifact needs quick refresh (rsQuickRefresh)."
End Select
```

ArtifactPersist.CanSave Method

Returns TRUE if the underlying object supports an explicit Save to storage function. Returns FALSE otherwise.

CanSave

Return Value

Returns TRUE if the underlying object supports an explicit Save to storage function. Returns FALSE otherwise.

Arguments

None

Example

```
Dim theArtifact As Artifact
If theArtifact.Persist.CanSave Then
   If theArtifact.Persist.NeedsSave Then
      PrintResults 0, "Artifact needs to be saved."
   Else
      PrintResults 0, "Artifact does not need to be saved."
   End If
   theArtifact.Persist.Save
Else
   PrintResults 0, "Artifact can't save."
End If
```

ArtifactPersist.NeedsRefresh Method

Returns rsNoRefresh if the object is known to be "up-to-date" relative to the current contents of persistent storage.

NeedsRefresh (eRefreshType bNeedsRefresh)

Return Value

Returns an eRefreshType. Returns rsNoRefresh if the object is known to be "up-to-date" relative to the current contents of persistent storage. Other values are rsNormalRefresh and rsQuickRefresh.

Arguments

None

Example

```
Dim theArtifact As Artifact
If theArtifact.Persist.CanRefresh() Then
   Select Case theArtifact.Persist.NeedsRefresh()
   Case rsNoRefresh
    PrintResults 0, "Artifact needs no refresh (rsNoRefresh)."
   Case rsNormalRefresh
    PrintResults 0, "Artifact needs normal refresh (rsNormalRefresh)."
   Case rsQuickRefresh
    PrintResults 0, "Artifact needs quick refresh (rsQuickRefresh)."
End Select
```

ArtifactPersist.NeedsSave Method

Returns FALSE if the contents of storage are known to be "up-to-date" with the object. Returns TRUE otherwise.

NeedsSave

Return Value

Returns FALSE if the contents of storage are known to be "up-to-date" with the object. Returns TRUE otherwise.

Arguments

None

Example

```
Dim theArtifact As Artifact
If theArtifact.Persist.CanSave Then
   If theArtifact.Persist.NeedsSave Then
      PrintResults 0, "Artifact needs to be saved."
   Else
      PrintResults 0, "Artifact does not need to be saved."
   End If
   theArtifact.Persist.Save
Else
   PrintResults 0, "Artifact can't save."
End If
```

ArtifactPersist.Refresh Method

Causes data in the current artifact to be refreshed from storage. This may be necessary if an external process modifies the data in the underlying persistent storage after the object has been read (that is, the current in memory copy of the object is stale relative to persistent storage). This can potentially be a very expensive operation depending on the capabilities of the underlying object.

Refresh

Return Value

None

Arguments

None

Example

```
If theArtifact.Persist.NeedsRefresh() <> rsNoRefresh Then
theArtifact.Persist.Refresh
End If
```

ArtifactPersist.Save Method

Causes data in the current artifact to be written out to persistent storage. Clients use this method to force data in persistent storage to be updated.

Save

Return Value

None

Arguments

None

Example

```
Dim theArtifact As Artifact
If theArtifact.Persist.CanSave Then
   If theArtifact.Persist.NeedsSave Then
      theArtifact.Persist.Save
Else
   PrintResults 0, "Artifact can't save."
End If
```

IArtifactProperty Interface

This is the default interface for the ArtifactProperty object. The ArtifactProperty object encapsulates the value associated with a property of a specific Artifact.

ArtifactProperty Properties

Table 31 lists the IArtifactProperty methods.

Table 31 IArtifactProperty Properties

Property	Description
Artifact	Returns the associated artifact object for this property.
Type	Returns the property type object associated with this property.
Value	Gets or sets an artifact property value.

ArtifactProperty.Artifact Property

Returns the associated Artifact object for this property.

Artifact

Return Value

Returns an Artifact object (IArtifact interface).

Arguments

None

Example

```
Dim theArtifactProperty As ArtifactProperty
Dim the Artifact As Artifact
theArtifact = theArtifactProperty.Artifact
```

ArtifactProperty.Type Property

Returns the property type object associated with this property.

Type

Return Value

Returns an ArtifactPropertyType object (IArtifactPropertyType interface).

Arguments

None

Example

```
Dim theProperty As ArtifactProperty
Dim theProperties As ArtifactPropertyCollection
Dim thePropertyType As ArtifactPropertyType

For PropID = 0 To theProperties.Count - 1
    Set theProperty = theProperties.Item(PropID)
    thePropertyType = theProperty.Type
Next PropID
```

ArtifactProperty.Value Property

Gets or sets the value of an ArtifactProperty. The value property is the mechanism for reading and writing artifact property values. The value data is passed as a VARIANT. The semantics of setting the value property is to actually modify the state of the associated artifact.

Value

Value(VARIANT newVal)

Return Value

Returns the value for the ArtifactProperty (IArtifactProperty interface).

Argument

newVal

A value as Variant for the ArtifactProperty.

Example

```
Private Sub Property_SetValue(theProperty As ArtifactProperty)
```

```
   Dim theOriginalValue As Variant
   Dim theNewValue As Variant
   Dim theNewValueCopy As Variant


   On Error GoTo ErrorTrap
   ' Get the current value of the property
   theOriginalValue = theProperty.Value
   theNewValue = GetVariantValueForPropertyType(theProperty.Type,
theOriginalValue)


   If theNewValue <> Empty Then
      theProperty.Value = theNewValue
      theNewValueCopy = theProperty.Value
      If (theNewValueCopy = theNewValue) Then
         PrintResults 2, "Original Value [" + CStr(theOriginalValue) +
"]; TestValue [" + CStr(theNewValue) + "]; Returned Value [" +
CStr(theNewValueCopy) + "]"
      Else
         PrintError 2, "Set " + CStr(theNewValue) + " but the object
returned " + CStr(theNewValueCopy)
      End If
      theProperty.Value = theOriginalValue
   End If


Exit Sub


ErrorTrap:
  PrintWarning 2, "Run-Error # " & CStr(Err.Number) & " " &
Err.Description
End Sub
```

IArtifactPropertyCollection Interface

This is the default interface for the collection of properties for an artifact. This interface encapsulates information about the list of artifact properties.

You create an artifact property collection object, in VB script, as follows:

```
Dim theSession
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating an Artifact Property Collection
Dim theArtifactPropertyCollection
' Set theArtifactPropertyCollection =
WScript.CreateObject("RDSICore.ArtifactPropertyCollection.1")
```

ArtifactPropertyCollection Properties

Table 32 lists the ArtifactPropertyCollection properties.

Table 32 ArtifactPropertyCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

ArtifactPropertyCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim thePropertyCollection As ArtifactPropertyCollection
Dim theCount As Integer
theCount = thePropertyCollection.Count
```

ArtifactPropertyCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns an ArtifactProperty object (IArtifactProperty interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim thePropertyCollection As ArtifactPropertyCollection
Dim theProperty As ArtifactProperty
Set theProperty = thePropertyCollection.Item(0)
```

ArtifactPropertyCollection Methods

Table 33 lists the ArtifactPropertyCollection methods.

Table 33 IArtifactPropertyCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

ArtifactPropertyCollection.Add Method

Appends the specified object to the end of the collection.

Add (IArtifactProperty Property)

Return Value

None

Argument

Property

An ArtifactProperty object (IArtifactProperty interface).

Example

```
Dim thePropertyCollection As ArtifactPropertyCollection
Dim theProperty As ArtifactProperty
Set thePropertyCollection = New ArtifactPropertyCollection
thePropertyCollection.Add theProperty
```

ArtifactPropertyCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IArtifactPropertyCollection Collection)

Return Value

None

Argument

Collection

An ArtifactPropertyCollection object (IArtifactPropertyCollection interface).

Example

```
Dim thePropertyCollection As ArtifactPropertyCollection
Dim thePropertyCollection2 As ArtifactPropertyCollection
Set thePropertyCollection = New ArtifactPropertyCollection
thePropertyCollection.AddCollection thePropertyCollection2
```

ArtifactPropertyCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim thePropertyCollection As ArtifactPropertyCollection
Dim theRetval As Boolean
theRetval = thePropertyCollection.IsModifiable
```

ArtifactPropertyCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim thePropertyCollection As ArtifactPropertyCollection
thePropertyCollection.Remove(0)
```

IArtifactPropertyType Interface

This is the default interface for the ArtifactPropertyType object.

ArtifactPropertyType Properties

Table 34 lists the ArtifactPropertyType properties.

Table 34 ArtifactPropertyType Properties

Property	Description
ArtifactType	Returns the artifact type object associated with this property type.
DataType	Returns the datatype for this property as a VARTYPE.
Key	Returns the Name property with the spaces and punctuation removed.
MaxSize	Returns the maximum size for this property in bytes.
Name	Returns the name of this property type.
PropertyID	Returns the PropertyID for this property type.
SemanticDataType	Returns the SemanticDataType for this property.

ArtifactPropertyType.ArtifactType Property

Returns the artifact type object associated with this property type.

ArtifactType

Return Value

Returns an ArtifactType object (IArtifactType interface)

Arguments

None

Example

```
Dim theArtifactType23 As ArtifactType
Dim theArtifactPropertyType23 As ArtifactPropertyType
```

```
Set theArtifactType23 = theArtifactPropertyType23.ArtifactType
```

ArtifactPropertyType.DataType Property

The DataType property represents the type of data (VarType) that is used by the variant value passed to and from IArtifactProperty.Value.

DataType

Return Value

Returns the datatype for this property as a VARTYPE.

Arguments

None

Example

```
Dim theArtifactPropertyType23 As ArtifactPropertyType
Dim theDataType23 As String
theDataType23 = theArtifactPropertyType23.DataType
```

ArtifactPropertyType.Key Property

Returns the key property value. The Key is the Name property with the spaces and punctuation removed.

Key

Return Value

Returns the Key property value.

Arguments

None

Example

```
Dim thePropertyType As ArtifactPropertyType
If thePropertyType.Name <> thePropertyType.Key Then
   NameText = NameText + " (" + thePropertyType.Key + ")"
End If
```

ArtifactPropertyType.MaxSize Property

Returns the maximum size for this property in bytes. If the DataType of this PropertyType is VT_BSTR, the property applies. If the MaxSize is -1, then the length is unlimited.

MaxSize

Return Value

Returns the maximum size for this property in bytes.

Arguments

None

Example

```
Dim theArtifactPropertyType As ArtifactPropertyType
If theArtifactPropertyType.MaxSize <> -1 Then
   TypeName = TypeName + "(Max Size = " + CStr(thePropertyType.MaxSize)
+ ")"
End If
```

ArtifactPropertyType.Name Property

Returns the name of this property type. The property type name is unique within an ArtifactType.

Name

Return Value

Returns the name of this property type.

Arguments

None

Example

```
Dim theArtifactPropertyType As ArtifactPropertyType
Dim thePropertyTypes As ArtifactPropertyCollection
Dim thePropertyType As ArtifactProperty
Set theArtifactPropertyType = thePropertyTypes.Item(PropID)
NameText = thePropertyType.Name
```

ArtifactPropertyType.PropertyID Property

Returns the PropertyID for this property type. This id is a unique number for this property. The number is unique within the scope of the given artifact type and all of its superclasses.

PropertyID

Return Value

Returns the PropertyID for this property type.

Arguments

None

Example

```
Dim thePropertyID As Long
Dim theArtifactPropertyType As ArtifactPropertyType
thePropertyID = theArtifactPropertyType.PropertyID
```

ArtifactPropertyType.SemanticDataType Property

Returns the SemanticDataType of this property as a SemanticDataType enumeration.

SemanticDataType

Return Value

Returns the semantic data type (eSemanticDataType) of this property type.

The semantic data types are: rsDataObject, rsDescription, rsDirectory, rsFileMoniker, rsFileOrDirectory, rsFilePath, rsName, rsNone, rsPassword, rsURL, rsUserName. The default is frwNone.

Arguments

None

Example

```
Dim thePropertyType As ArtifactPropertyType
Dim theSemanticDataType As SemanticDataType
theSemanticDataType = thePropertyType.SemanticDataType
Select Case theSemanticDataType
```

```
  Case rsDirectory
    GetSemanticTypeName = "Directory"
  Case rsFileOrDirectory
    GetSemanticTypeName = "FileOrDirectory"
  Case rsFilePath
    GetSemanticTypeName = "FilePath"
  Case rsName
    GetSemanticTypeName = "Name"
  Case rsUserName
    GetSemanticTypeName = "UserName"
  Case rsPassword
    GetSemanticTypeName = "Password"
End Select
```

ArtifactPropertyType Methods

Table 35 lists the IArtifactPropertyType methods.

Table 35 IArtifactPropertyType Methods

Method	Description
IsDynamicType	Returns TRUE if this instance is a dynamic type.
SetAllowed	Returns TRUE if the properties of this type can be set.

ArtifactPropertyType.IsDynamicType Method

Returns TRUE if this instance is a dynamic type. A dynamic type is defined as a type that is specific to a given instance of an artifact. Static types are always available.

IsDynamicType

Return Value

Returns TRUE if this instance is a dynamic type, FALSE otherwise.

Arguments

None

Example

```
Dim theArtifactPropertyType As ArtifactPropertyType
Dim DynamicText As String
If theArtifactPropertyType.IsDynamicType() = True Then
   DynamicText = "Dynamic"
Else
   DynamicText = "Static"
End If
```

ArtifactPropertyType.SetAllowed Method

Returns TRUE if the properties of this type can be set. Returns False if the property can not be set.

SetAllowed

Return Value

Returns TRUE if the property can be set. Returns FALSE if the property is read-only.

Arguments

None

Example

```
Dim thePropertyType As ArtifactPropertyType
If thePropertyType.SetAllowed() Then
   ModText = JustifyString("Yes", ModWidth, True)
Else
   ModText = JustifyString("No", ModWidth, True)
End If
```

IArtifactPropertyTypeCollection Interface

This is the default interface for the collection of property types for an artifact. This interface encapsulates information about the list of artifact property types.

ArtifactPropertyTypeCollection Properties

Table 36 lists the ArtifactPropertyTypeCollection properties.

Table 36 ArtifactPropertyTypeCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

ArtifactPropertyTypeCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim thePropertyTypeCollection As ArtifactPropertyTypeCollection
Dim theCount As Integer
theCount = thePropertyTypeCollection.Count
```

ArtifactPropertyTypeCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns an ArtifactPropertyType object (IArtifactPropertyType interface).

Arguments

index

A variant index to an item in the collection.

Example

```
Dim thePropertyTypeCollection As ArtifactPropertyTypeCollection
Dim thePropertyType As ArtifactPropertyType
Set thePropertyType = thePropertyTypeCollection.Item(0)
```

ArtifactPropertyTypeCollection Methods

Table 37 lists the ArtifactPropertyTypeCollection methods.

Table 37 IArtifactPropertyTypeCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

ArtifactPropertyTypeCollection.Add Method

Appends the specified object to the end of the collection.

Add (IArtifactPropertyType Property)

Return Value

None

Argument

Property

An ArtifactPropertyType object (IArtifactPropertyType interface)

Example

```
Dim thePropertyTypeCollection As ArtifactPropertyTypeCollection
Dim thePropertyType As ArtifactPropertyType
Set thePropertyTypeCollection = New ArtifactPropertyTypeCollection
thePropertyTypeCollection.Add thePropertyType
```

ArtifactPropertyTypeCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IArtifactPropertyTypeCollection Collection)

Return Value

None

Argument

Collection

An ArtifactPropertyTypeCollection object (IArtifactPropertyTypeCollection interface).

Example

```
Dim thePropertyTypeCollection As ArtifactPropertyTypeCollection
Dim thePropertyTypeCollection2 As ArtifactPropertyTypeCollection
Set thePropertyTypeCollection = New ArtifactPropertyTypeCollection
thePropertyTypeCollection.AddCollection thePropertyTypeCollection2
```

ArtifactPropertyTypeCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim thePropertyTypeCollection As ArtifactPropertyTypeCollection
Dim theRetval As Boolean
theRetval = thePropertyTypeCollection.IsModifiable
```

ArtifactPropertyTypeCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim thePropertyTypeCollection As ArtifactPropertyTypeCollection
thePropertyTypeCollection.Remove(0)
```

IArtifactType Interface

This is the default interface for the ArtifactType object.

In addition to artifact type support, this interface includes:

- Property Support

  PropertyTypes

- Relationship Support

  GetRelatedTypes

  GetRelationshipsOfType

  GetRelationshipTypes

- Superclass/Subclass support

  SuperClass

  SubClasses

- Graphics Support

  GraphicsFormats

ArtifactType Properties

Table 38 lists the ArtifactType properties.

Table 38 ArtifactType Properties

Property	Description
ClassID	Returns the ClassID for this artifact type.
GraphicsFormat	Support for rendering graphics to files. If the artifact type supports rendering, then there will be objects contained in this collection.
Key	Returns the key for the artifact type.
LocatorTypes	Returns the LocatorTypes registered for this Artifact Type.
MethodTypes	Returns the collection of method types defined by this artifact type.

Property	Description
Name	Returns the name of the artifact type.
PropertyTypes	Returns the collection of property types defined by this artifact type.
SubClasses	Returns all artifact types that derive from this artifact type.
SuperClass	Returns the superclass of the artifact.

ArtifactType.ClassID Property

Returns the ClassID for this artifact type. ClassID is a unique id for this artifact type.

ClassID

Return Value

Returns the ClassID for this artifact type.

Arguments

None

Example

```
Dim theArtifactType As ArtifactType
Dim theClassID As Long
theClassID = theArtifactType.ClassID
```

ArtifactType.GraphicsFormats Property

Support for rendering graphics to files. If the artifact type supports rendering, then there will be objects contained in this collection.

GraphicsFormats

Return Value

Returns an ArtifactGraphicsFormatTypeCollection object (IArtifactGraphicsFormatTypeCollection interface).

Arguments

None

Example

```
Dim GraphicsFormats As ArtifactGraphicsFormatTypeCollection
Dim GraphicsFormat As ArtifactGraphicsFormatType
Dim FormatID As Integer
Dim Results As String
Set GraphicsFormats = m_ContextArtifact.Type.GraphicsFormats
If GraphicsFormats.Count = 0 Then
   PrintComment 1, "This object does not support rendering graphics."
   Exit Function
End If
For FormatID = 0 To GraphicsFormats.Count - 1
   Set GraphicsFormat = GraphicsFormats.Item(FormatID)
   PrintResults 1, "Graphics Format [Name = " + GraphicsFormat.Name +
"]"
   PrintResults 1, " [Description = " + GraphicsFormat.Description +
"]"
   PrintResults 1, " [Format = " + GraphicsFormat.Format + "]"
   Dim Ext As String
   Ext = GraphicsFormat.Format
   theArtifact.RenderToFile GraphicsFormat, "File." + Ext
   Results = theArtifact.RenderToFileEx(GraphicsFormat,
"File(Path='FileEx." + Ext + "')|Format(Width=1 Height=1
UnitsPerInch=1)")
   PrintResults 1, "RenderToFileEx results: [" + Results + "]"
Next FormatID
```

ArtifactType.Key Property

Returns the key for the artifact type. IArtifactType.Key contains the Name property with the spaces and punctuation removed.

Key

Return Value

Returns the key for the artifact type.

Arguments

None

Example

```
Dim theArtifactType As ArtifactType
Dim NameText As String
NameText = theArtifactType.Name
If theArtifactType.Name <> theArtifactType.Key Then
   NameText = NameText + " (" + theArtifactType.Key + ")"
End If
```

ArtifactType.LocatorTypes Property

Returns the locator types registered for this Artifact Type.

LocatorTypes

Return Value

Returns the ArtifactLocatorTypeCollection object containing the registered locator types for this Artifact Type (IArtifactLocatorTypeCollection interface).

Arguments

None

Example

```
Dim m_ContextArtifact As Artifact
Dim theLocatorTypes As ArtifactLocatorTypeCollection
Set theLocatorTypes = m_ContextArtifact.Type.MethodTypes
```

ArtifactType.MethodTypes Property

Returns the collection of method types defined by this artifact type.

MethodTypes

Return Value

Returns the MethodTypeCollection object containing the registered method types for this Artifact Type (IMethodTypeCollection interface).

Arguments

None

Example

```
Dim m_ContextArtifact As Artifact
Dim theMethodTypes As MethodTypeCollection
 Set theMethodTypes = m_ContextArtifact.Type.MethodTypes
```

ArtifactType.Name Property

Returns the name of the artifact type. The name property uniquely identifies an ArtifactType within a provider.

Name

Return Value

Returns the name of the artifact type.

Arguments

None

Example

```
Dim theArtifactType As ArtifactType
Dim NameText As String
NameText = theArtifactType.Name
```

ArtifactType.PropertyTypes Property

Returns the collection of property types defined by this artifact type.

PropertyTypes

Return Value

Returns an ArtifactPropertyTypeCollection object (IArtifactPropertyTypeCollection interface).

Arguments

None

Example

```
Dim theArtifactType As ArtifactType
Dim thePropertyTypes As ArtifactPropertyTypeCollection
Set thePropertyTypes = theArtifactType.PropertyTypes
```

ArtifactType.SubClasses Property

Returns all artifact types that derive from this. Derived types inherit all properties, child types, and relationship types from this artifact type.

SubClasses

Return Value

Returns an ArtifactTypeCollection object (IArtifactTypeCollection interface).

Arguments

None

Example

```
Dim theSubclasses As ArtifactTypeCollection
Dim theArtifactType As ArtifactType
Set theSubclasses = theArtifactType.SubClasses
```

ArtifactType.SuperClass Property

Returns the superclass of the artifact. An artifact inherits its properties and relationships from its superclass.

SuperClass

Return Value

Returns an ArtifactType object (IArtifactType interface).

Arguments

None

Example

```
Dim theArtifactType As ArtifactType
Set theArtifactType = theArtifactType.SuperClass
If Not theArtifactType Is Nothing Then
   PrintResults IndentLevel, theArtifactType.Name + ": " +
theArtifactType.SuperClass.Name
Else
   PrintResults IndentLevel, theArtifactType.Name + " (no superclass)"
```

```
End If
```

ArtifactType Methods

Table 39 lists the IArtifactType methods.

Table 39 IArtifactType Methods

Method	Description
CreateLocator	Returns the default locator for the given artifact type.
CreateLocatorEx	Creates a locator using the specific locator type.
CreateLocatorEx_ID	Creates a locator using the specific locator type id.
GetRelatedTypes	Returns the collection of Artifact Types that are related to this type.
GetRelationshipsOfType	Returns the collection of relationships of a specific type.
GetRelationshipTypes	Returns the collection of relationship types that are valid for this artifact.
IsAbstractType	Returns TRUE if this instance is an abstract type. Returns FALSE otherwise.
IsDescendentType	Returns TRUE if the specified name is a legal descendent type for this type. Returns FALSE otherwise.
IsDynamicType	Returns TRUE if this instance is a dynamic type.

ArtifactType.CreateLocator Method

Returns the default locator for the given artifact type.

CreateLocator (eLocatorType LocatorType)

Return Value

Returns an ArtifactLocator object (IArtifactLocator interface).

Arguments

LocatorType

An eLocatorType. The locator types are: rs_Locator_Display_Name, rs_Locator_Immutable_ID, rs_Locator_Unspecified.

Example

```
Dim theArtifact As Artifact
Dim theArtifactType As ArtifactType
Dim theLocator As ArtifactLocator
Set theArtifactType = theArtifact.Type
Set theLocator = theArtifactType.CreateLocator(rsLocator_Display_Name)
```

ArtifactType.CreateLocatorEx Method

Creates a locator using the specific locator type. The argument values are un-initialized. If a relative locator is passed, then this locator is used. However, the artifact type of the relative locator must match the relative artifact type of theLocatorType. If no relative locator is passed and one is needed, then an appropriate default locator for the relative type is chosen.

CreateLocatorEx (IArtifactLocatorType LocatorType, IArtifactLocator RelativeLocator)

Return Value

Returns a reference to an ArtifactLocator object (IArtifactLocator interface).

Arguments

LocatorType

Specifies the locator type to create.

RelativeLocator

An optional argument for specifying a relative locator. The ArtifactType of the RelativeLocator must match the relative artifact type of the LocatorType.

Example

```
Dim theArtifact As Artifact
Dim theArtifactType As ArtifactType
Dim theLocator As ArtifactLocator
Dim theLocatorType As ArtifactLocatorType
Set theArtifactType = theArtifact.Type
Set theLocator = theArtifactType.CreateLocatorEx(theLocatorType)
```

ArtifactType.CreateLocatorEx_ID Method

CreateLocatorEx_ID is intended for custom purposes. See CreateLocatorEx.

Creates a locator using the specific locator type id. Otherwise, identical to CreateLocatorEx.

Creates a locator for the artifact using the specific locator type. The locator is initialized with the argument values specific to the artifact. If a relative locator is passed, then this locator is used. However, the artifact type of the relative locator must match the relative artifact type of theLocatorType. If no relative locator is passed and one is needed, then an appropriate default locator for the relative type is chosen.

Note: Methods using internal ids are intended to be called by generated product-specific wrapper classes.

CreateLocatorEx_ID (LONG ClassID, LONG ArtifactLocatorTypeID, IArtifactLocator RelativeLocator)

Return Value

Returns an ArtifactLocator object (IArtifactLocator interface).

Arguments

ClassID

A unique id for the ArtifactType of the locator to create.

ArtifactLocatorTypeID

Specifies the unique id of the locator type to create.

RelativeLocator

An optional argument for specifying a relative locator. The ArtifactType of the RelativeLocator must match relative artifact type of the LocatorType.

Example

```
Dim theArtifact As Artifact
Dim theArtifactType As ArtifactType
Dim theLocator As ArtifactLocator
Dim theLocatorType As ArtifactLocatorType
Set theArtifactType = theArtifact.Type
‘Get the ClassID
```

```
Dim theClassID As Long
theClassID = theArtifactType.ClassID
‘Get the LocatorTypeID
Dim theLocatorTypeID As String
theLocatorTypeID = theLocatorType.LocatorTypeID
Set theLocator = theArtifactType.CreateLocatorEx_ID(theClassID,
theLocatorTypeID)
```

ArtifactType.GetRelatedTypes Method

Returns the collection of artifact types that are related to this type.

GetRelatedTypes (eRelationshipCategory Category)

Return Value

Returns an ArtifactTypeCollection object (IArtifactTypeCollection interface).

Argument

Category

An eRelationshipCategory (for example, rsAll, rsDescendant, rsPeer, or rsChild).

Example

```
Dim theArtifact As Artifact
Dim theArtifactTypeCollection As ArtifactTypeCollection
Set theArtifactTypeCollection = theArtifact.Type.GetRelatedTypes(peer)
```

ArtifactType.GetRelationshipsOfType Method

Returns the collection of relationships of a specific type.

GetRelationshipsOfType (eRelationshipCategory Category, IArtifactType ArtifactType, BOOL IncludeUnspecifiedTypes)

Return Value

Returns a RelationshipTypeCollection object (IRelationshipTypeCollection interface).

Arguments

Category

An eRelationshipCategory (for example, rsAll, rsDescendant, rsPeer, or rsChild).

ArtifactType

An ArtifactType object (IArtifactType interface).

IncludeUnspecifiedTypes

A boolean optional argument. If set to TRUE, this method returns all relationship types for the given artifact type, including all unspecified relationship types.

Example

```
Dim theArtifactTypeCollection As ArtifactTypeCollection
Dim theArtifactType As ArtifactType
Dim theRelatedArtifactType As ArtifactType
Dim ArtifactTypeID As Integer
For ArtifactTypeID = 0 To theArtifactTypeCollection.Count - 1
   Set theRelatedArtifactType =
theArtifactTypeCollection.Item(ArtifactTypeID)
   Set theRelationships =
theArtifactType.GetRelationshipsOfType(RelCategory,
theRelatedArtifactType)
...
Next ArtifactTypeID
```

ArtifactType.GetRelationshipTypes Method

Returns the collection of relationship types that are valid for this artifact. Returns an empty collection if this artifact type does not support relationships. Allows either Child, Descendant, or Peer relationships to be returned.

GetRelationshipTypes (eRelationshipCategory Category)

Return Value

Returns a RelationshipTypeCollection object (IRelationshipTypeCollection interface).

Argument

Category

A relationship category.

Example

```
Dim theArtifactType As ArtifactType
Dim theRelationships As RelationshipTypeCollection
```

```
Set theRelationships = theArtifactType.GetRelationshipTypes(peer)
```

ArtifactType.IsAbstractType Method

Returns TRUE if this instance is an abstract type. Returns FALSE otherwise.

There are some artifact types registered by adapters that can never be instantiated. They are used to define subclass artifact types. Examples of these are Rose Item, FileSys DirectoryObject and ClearCase VobObject.

IsAbstractType

Return Value

Returns TRUE if this instance is an abstract type. Returns FALSE otherwise.

Arguments

None

Example

```
Dim theArtifact As Artifact
Dim theArtifactType As ArtifactType
Dim theLocator As ArtifactLocator
Dim theText As String
Set theArtifactType = theArtifact.Type
If theArtifactType.IsAbstractType() = True Then
   theText = "Abstract"
Else
Set theLocator = theArtifactType.CreateLocator(rsLocator_Display_Name)
End If
```

ArtifactType.IsDescendentType Method

Tests for the child relationship. Returns TRUE if the specified name is a legal descendent type for this type. Returns FALSE otherwise.

IsDescendentType (BSTR name)

Return Value

Returns TRUE if the specified name is a legal descendent type for this type.

Argument

name

The name of the artifact type to check.

Example

```
Dim theSubclass As ArtifactType
Dim theArtifactType As ArtifactType
Dim theLocator As ArtifactLocator
If theArtifactType.IsDescendentType(theSubclass.Name) = False Then
   PrintError IndentLevel + 1, "theArtifactType.IsDescendentType (" &
theSubclass.Name & ") failed!"
End If
```

ArtifactType.IsDynamicType Method

Returns TRUE if this instance is a dynamic type. A dynamic type is defined as a type that is specific to a given instance of an artifact. Static types are always available.

IsDynamicType

Return Value

Returns TRUE if this instance is a dynamic type. Returns FALSE otherwise.

Arguments

None

Example

```
Dim theArtifactType As ArtifactType
Dim DynamicText As String
If theArtifactType.IsDynamicType() = True Then
   DynamicText = "Dynamic"
Else
   DynamicText = "Static"
End If
```

IArtifactTypeCollection Interface

This is the default interface for the collection of artifact types for an adapter or an artifact. This interface encapsulates information about the list of artifact types. This includes static artifact types for an adapter and both static and dynamic types for an artifact.

You create an artifact type collection object, in VB script, as follows:

```
Dim theSession
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating an Artifact Type Collection
Dim theArtifactTypeCollection
Set theArtifactTypeCollection =
WScript.CreateObject("RDSICore.ArtifactTypeCollection.1")
```

ArtifactTypeCollection Properties

Table 40 lists the ArtifactTypeCollection properties.

Table 40 ArtifactTypeCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

ArtifactTypeCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim theCount26 As Integer
Dim theArtifactTypeCollection As ArtifactTypeCollection
theCount26 = theArtifactTypeCollection.Count
```

ArtifactTypeCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns an artifact type object (IArtifactType interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theItem As ArtifactType
Dim theArtifactTypeCollection As ArtifactTypeCollection
Set theItem = theArtifactTypeCollection.Item(1)
```

ArtifactTypeCollection Methods

Table 41 lists the ArtifactTypeCollection methods.

Table 41 IArtifactTypeCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

ArtifactTypeCollection.Add Method

Appends the specified object to the end of the collection.

Add (IArtifactType ArtifactType)

Return Value

None

Argument

ArtifactType

An ArtifactType object (IArtifactType interface).

Example

```
Dim theArtifactTypeCollection As ArtifactTypeCollection
Dim theArtifactType As ArtifactType
Dim theArtifact As Artifact
Set theArtifactType = theArtifact.Type
theArtifactTypeCollection.Add theArtifactType
```

ArtifactTypeCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IArtifactTypeCollection Collection)

Return Value

None

Argument

Collection

An ArtifactTypeCollection object (IArtifactTypeCollection interface).

Example

```
Dim theArtifactType26 As ArtifactType
Dim theArtifact26 As Artifact
Dim theArtifactTypeCollection26 As ArtifactTypeCollection
Set theArtifactType26 = theArtifact26.Type
```

```
Set theArtifactTypeCollection26 =
theArtifactType26.GetRelatedTypes(peer)


Dim theArtifactTypeCollection261 As ArtifactTypeCollection
Set theArtifactTypeCollection261 =
theArtifactType26.GetRelatedTypes(Child)
theArtifactTypeCollection26.AddCollection theArtifactTypeCollection261
```

ArtifactTypeCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theMod26 As Boolean
Dim theArtifactTypeCollection As ArtifactTypeCollection
theMod26 = theArtifactTypeCollection.IsModifiable
```

ArtifactTypeCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim theArtifactTypeCollection As ArtifactTypeCollection
theArtifactTypeCollection.Remove (0)
```

IFilterComparison Interface

This interface contains a simple comparison expression, including two operands and one operator.

FilterComparison Properties

Table 42 lists the IFilterComparison properties.

Table 42 IFilterComparison Properties

Property	Description
ComparisonOp	Returns the comparison operator for the filter comparison.
PropertyName	Returns the property name for the filter comparison.
Value	Returns the value for the filter comparison.

FilterComparison.ComparisonOp Property

Returns the comparison operator for the filter.

ComparisonOp

Return Value

Returns a comparison operator.

Arguments

None

Example

```
Dim theComparison As FilterComparison
Select Case theComparison.ComparisonOp
Case COMP_OP_EQ
  SOp = " = "
Case COMP_OP_BETWEEN
  SOp = " BETWEEN "
Case COMP_OP_GT
```

```
  SOp = " > "
Case COMP_OP_GTE
  SOp = " >= "
Case COMP_OP_IN
  SOp = " IN "
Case COMP_OP_LIKE
  SOp = " LIKE "
Case COMP_OP_LT
  SOp = " < "
Case COMP_OP_LTE
  SOp = " <= "
Case COMP_OP_NEQ
  SOp = " != "
Case COMP_OP_NOT_BETWEEN
  SOp = " NOT BETWEEN "
Case COMP_OP_NOT_IN
  SOp = " NOT IN "
Case COMP_OP_NOT_LIKE
  SOp = " NOT LIKE "
Case COMP_OP_IS_KIND_OF
  SOp = " IsKindOf "
End Select
```

FilterComparison.PropertyName Property

Returns the property name for the filter.

PropertyName

Return Value

Returns the property name for the filter.

Arguments

None

Example

```
Dim theComparison As FilterComparison
theComparisonName As String
```

```
theComparisonName = theComparison.Name
```

FilterComparison.Value Property

Returns the value for the filter comparison.

Value

Return Value

Returns the value for the filter comparison.

Arguments

None

Example

```
Dim theComparison As FilterComparison
theComparisonValue As Variant
theComparisonValue = theComparison.Value
```

IFilterComparisonCollection Interface

A collection of filter comparisons. No modification of the collection is needed.

You create a filter comparison collection object, in VB script, as follows:

```
Dim theSession
' Creating an RDSI Session
Set theSession = WScript.CreateObject("RDSICore.Session.1")
' Creating a Filter Comparison Collection
Dim theFilterComparisonCollection
' Set theFilterComparisonCollection =
WScript.CreateObject("RDSICore.FilterComparisonCollection.1")
```

FilterComparisonCollection Properties

Table 43 lists the FilterComparisonCollection properties.

Table 43 FilterComparisonCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

FilterComparisonCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim theComparisonCollection As FilterComparisonCollection
```

```
Dim theCount26 As Integer
theCount26 = theComparisonCollection.Count
```

FilterComparisonCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns a FilterComparison object (IFilterComparison interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theItem As FilterComparison
Dim theComparisonCollection As FilterComparisonCollection
Set theItem = theComparisonCollection.Item(1)
```

FilterComparisonCollection Methods

Table 44 lists the FilterComparisonCollection methods.

Table 44 IFilterComparisonCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

FilterComparisonCollection.Add Method

Appends the specified object to the end of the collection.

Add (IFilterComparison FilterComparison)

Return Value

None

Argument

FilterComparison

Returns a FilterComparison object (IFilterComparison interface).

Example

```
Dim theComparison As FilterComparison
Dim theComparisonCollection As FilterComparisonCollection
Set theComparisonCollection = New FilterComparisonCollection
theComparisonCollection.Add theComparison
```

FilterComparisonCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IFilterComparisonCollection Collection)

Return Value

None

Argument

Collection

A FilterComparisonCollection object (IFilterComparisonCollection interface).

Example

```
Dim theComparison As FilterComparison
Dim theComparisonCollection As FilterComparisonCollection
Dim theComparisonCollection2 As FilterComparisonCollection

Set theComparisonCollection = New FilterComparisonCollection
```

```
theComparisonCollection.AddCollection theComparisonCollection2
```

FilterComparisonCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theMod26 As Boolean
Dim theComparisonCollection As FilterComparisonCollection
theMod26 = theComparisonCollection.IsModifiable
```

FilterComparisonCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim theComparisonCollection As FilterComparisonCollection
theComparisonCollection.Remove(0)
```

IMethod Interface

This is the default interface an artifact method.

Note: This interface is currently not supported.

Method Properties

Table 45 lists the Method properties.

Table 45 Method Properties

Property	Description
Arguments	Returns the collection of arguments that apply to this method.
Artifact	Returns the associated Artifact object for this method.
MethodType	Returns the MethodType of this Method.

Method.Arguments Property

Returns the collection of arguments that apply to this method.

Arguments

Return Value

Returns the ArtifactArgumentCollection object that applies to this method (IArtifactArgumentCollection interface).

Arguments

None

Example

```
Dim theMethodType As MethodType
Dim theMethod As Method
Dim m_ContextArtifact As Artifact
Dim theArguments As ArtifactArgumentCollection
Dim theReturnArgument As ArtifactArgument
```

```
Dim theMethodTypes As MethodTypeCollection

Set theMethodTypes = m_ContextArtifact.Type.MethodTypes
Set theMethodType = theMethodTypes.Item("Revert")
Set theMethod = m_ContextArtifact.CreateMethod(theMethodType)
Set theArguments = theMethod.Arguments
```

Method.Artifact Property

Returns the associated Artifact object for this method.

Artifact

Return Value

Returns an Artifact object (IArtifact interface).

Arguments

None

Example

```
Dim theArtifact As Artifact
Dim theMethodType As MethodType
Dim theMethod As Method
Set theArtifact = theMethod.Artifact
```

Method.MethodType Property

Returns the MethodType of this Method.

MethodType

Return Value

Returns the MethodType of this Method (IMethodType interface).

Arguments

None

Example

```
Dim theMethodType As MethodType
```

```
Dim theMethod As Method
Set theMethodType = theMethod.MethodType
```

Method Interface Methods

Method.Execute Method

Executes the method. Its return value is the method's return value, stored in an ArtifactArgument. You need to call getValue() on that argument in order to find out what was returned.

Execute

Return Value

Returns a reference to the ArtifactArgument object that applies to this method (IArtifactArgument interface).

Arguments

None

Example

```
Dim theMethodType As MethodType
Dim theMethod As Method
Dim m_ContextArtifact As Artifact
Dim theArguments As ArtifactArgumentCollection
Dim theReturnArgument As ArtifactArgument
Dim theMethodTypes As MethodTypeCollection
Set theMethodTypes = m_ContextArtifact.Type.MethodTypes
Set theMethodType = theMethodTypes.Item("Save")
Set theMethod = m_ContextArtifact.CreateMethod(theMethodType)
Set theReturnArgument = theMethod.Execute
```

IMethodType Interface

This is the default interface for an artifact method type.

Note: This interface is currently not supported.

MethodType Properties

Table 46 lists the MethodType properties.

Table 46 MethodType Properties

Property	Description
Arguments	Returns the collection of arguments that apply to this method type.
ArtifactType	Returns the ArtifactType object associated with this method type.
MethodTypeID	Returns the unique id for the method type used for indexed lookup.
Name	Returns the name of the artifact type associated with this method type.
ReturnValue	Returns the argument that describes the return value.

MethodType.Arguments Property

Returns the collection of arguments that apply to this method type.

Arguments

Return Value

Returns the ArtifactArgumentCollection object that applies to this method type (IArtifactArgumentCollection interface).

Arguments

None

Example

```
Dim theArguments As ArtifactArgumentCollection
Dim theMethodType As MethodType
Set theArguments = theMethodType.Arguments
```

MethodType.ArtifactType Property

Returns the ArtifactType object associated with this method type (IArtifactType interface).

ArtifactType

Return Value

Returns the ArtifactArgumentCollection object that applies to this method (IArtifactArgumentCollection interface).

Arguments

None

Example

```
Dim theArtifactType As ArtifactType
Dim theMethodType As MethodType
Set theArtifactType = theMethodType.ArtifactType
```

MethodType.MethodTypeID Property

Returns the unique ID for the method type used for indexed lookup.

MethodTypeID

Return Value

Returns unique ID for the method type.

Arguments

None

Example

```
Dim theMethodID As String
Dim theMethodType As MethodType
```

```
Set theMethodID = theMethodType.MethodTypeID
```

MethodType.Name Property

Returns the name of the artifact type associated with this method type. This represents the descriptive name of the method.

Name

Return Value

Returns the name of the artifact type associated with this method type.

Arguments

None

Example

```
Dim theMethodName As String
Dim theMethodType As MethodType
Set theMethodName = theMethodType.Name
```

MethodType.ReturnValue Property

Returns the argument that describes the return value.

ReturnValue

Return Value

Returns the ArtifactArgument object that describes the return value. (IArtifactArgument interface).

Arguments

None

Example

```
Dim theMethodType As MethodType
Dim theMethod As Method
Dim index As Integer
Dim theReturnArgument As ArtifactArgument
For index = 0 To theMethodTypes.Count - 1
Set theMethodType = theMethodTypes.Item(index)
```

```
Set theReturnArgument = theMethodType.ReturnValue
If theReturnArgument Is Nothing Then
   ‘business logic
   Else
   ‘other business logic
End If
Next index
```

IMethodTypeCollection Interface

This is the default interface for the collection of method types. This interface encapsulates information about the list of method types for an artifact.

Note: This interface is currently not supported.

MethodTypeCollection Properties

Table 47 lists the MethodTypeCollection properties.

Table 47 MethodTypeCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

MethodTypeCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim theMethodTypeCollection As MethodTypeCollection
Dim theCount As Integer
theCount = theMethodTypeCollection.Count
```

MethodTypeCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns a MethodType object (IMethodType interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theMethodTypeCollection As MethodTypeCollection
Dim theItem As MethodType
Set theItem = theMethodTypeCollection.Item(1)
```

MethodTypeCollection Methods

Table 48 lists the FilterComparisonCollection methods.

Table 48 MethodTypeCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

MethodTypeCollection.Add Method

Appends the specified object to the end of the collection.

Add (IMethodType MethodType)

Return Value

None

Argument

MethodType

A MethodType object (IMethodType interface).

Example

```
Dim theMethodTypeCollection As MethodTypeCollection
Dim theMethodType As MethodType
Set theMethodTypeCollection = New MethodTypeCollection
theMethodTypeCollection.Add theMethodType
```

MethodTypeCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IMethodTypeCollection Collection)

Return Value

None

Argument

Collection

A MethodTypeCollection object (IMethodTypeCollection interface).

Example

```
Dim theMethodTypeCollection As MethodTypeCollection
Dim theMethodTypeCollection2 As MethodTypeCollection
Set theMethodTypeCollection = New MethodTypeCollection
theMethodTypeCollection.AddCollection theMethodTypeCollection2
```

MethodTypeCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise (FALSE if the collection is owned by an object). In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theMethodTypeCollection As MethodTypeCollection
Dim theModifiableProperty As Boolean
theModifiableProperty = theMethodTypeCollection.IsModifiable
```

MethodTypeCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

```
Dim theMethodTypeCollection As MethodTypeCollection
theMethodTypeCollection.Remove(0)
```

IRDSISession Interface

This is the default interface of the RDSISession object.

The session object is the point of entry into the RSE interfaces. You begin by creating an RDSISession. For example, in VB script:

```
Dim theSession
Set theSession = WScript.CreateObject("RDSICore.Session.1")
```

Once you create a new Session, you can use the RDSISession methods to retrieve the registered adapters, create an artifact locator, and locate artifacts.

RDSISession Properties

Table 49 lists the RDSISession properties.

Table 49 RDSISession Properties

Property	Description
Adapters	Returns the collection of registered adapters

RDSISession.Adapters Property

Returns the collection of adapters that are registered with the RSE core on this system.

Adapters

Return Value

Returns an AdapterCollection object (IAdapterCollection interface).

Arguments

None

Example

```
Dim theSession As RDSISession
Dim theAdapter As Adapter
Set theSession = GetRDSISession()
Set theAdapter = theSession.Adapters.Item(AdapterName)
```

```
Public Function GetRDSISession() As RDSISession
Dim theRDSISession As RDSISession
Set GetRDSISession = New RDSISession
End Function
```

VB Script Example

```
Dim theSession,theTestAdapter
Set theTestAdapter = theSession.Adapters.Item(2)
Dim adapName
adapName = theTestAdapter.Name
```

RDSISession Methods

Table 50 lists the RDSISession methods.

Table 50 IRDSISession Methods

Method	Description
CreateArtifactLocator	Returns a locator object
LocateArtifact	Returns an artifact

RDSISession.CreateArtifactLocator Method

Returns an artifact locator (IArtifactLocator interface), given an ArtifactID string.

CreateArtifactLocator (BSTR ArtifactID)

Return Value

Returns an ArtifactLocator object (IArtifactLocator interface).

Argument

ArtifactID

The locator string associated with the artifact.

Example

```
   SetArtifactID(ArtifactID As String)
Dim theLocator As ArtifactLocator
Dim theSession As RDSISession
```

```
  Set theSession = GetRDSISession()
  Set theLocator = theSession.CreateArtifactLocator(ArtifactID)

Public Function GetRDSISession() As RDSISession
Dim theRDSISession As RDSISession
Set GetRDSISession = New RDSISession
End Function
```

VB Script Example

```
Dim theSession, theLocator
Set theSession = WScript.CreateObject("RDSICore.Session.1")
‘Create a Locator for a ReqPro Project
Set theLocator =
theSession.CreateArtifactLocator("ReqPro|Project(Path='D:\Program
Files\Rational\Requisitepro\samples\Learning_Project\Business\Learning
_Business.RQS')")
```

RDSISession.LocateArtifact Method

Given an ArtifactID, use this method to locate an artifact from the session object. Returns the IArtifact for the Artifact object identified by the specified ArtifactID string.

LocateArtifact (BSTR ArtifactID)

Return Value

Returns an Artifact object (IArtifact interface).

Argument

ArtifactID

ArtifactID is the locator string associated with the artifact.

Example

```
Dim theRDSISession As RDSISession
Dim theAdapter As Adapter
Set theRDSISession = GetRDSISession()
Set theAdapter = GetCurrentAdapter()
Set theArtifactTypes = theAdapter.LocateArtifact(GetArtifactID)

Public Function GetRDSISession() As RDSISession
```

```
Dim theRDSISession3 As RDSISession
Set GetRDSISession = New RDSISession
End Function
Public Function GetCurrentAdapter(AdapterID As Integer, theRDSISession
As RDSISession) As Adapter
Dim theAdapter As Adapter
Set GetCurrentAdapter = theRDSISession.Adapters.Item(AdapterID)
End Function
```

VB Script Example

```
Dim theSession, theArtifact
Set theSession = WScript.CreateObject("RDSICore.Session.1")
‘Locate a ReqPro Project
Set theArtifact =
theSession.LocateArtifact("ReqPro|Project(Path='D:\Program
Files\Rational\Requisitepro\samples\Learning_Project\Business\Learning
_Business.RQS')")
```

IRelationshipType Interface

This is the default interface for RelationshipType objects.

RelationshipType Properties

Table 51 lists the RelationshipType properties.

Table 51 RelationshipType Properties

Property	Description
ArtifactType	Returns the ArtifactType object associated with this relationship type.
Cardinality	Returns either NARY or UNARY (many or one).
Category	Returns a relationship type category.
CreationArguments	Returns an ArtifactArguments collection suitable for passing to IArtifact.CreateArtifact.
Key	Returns the key of this relationship type.
Name	Returns the name of this relationship type.
RelationshipID	Returns the RelationshipID of the relationship type.

RelationshipType.ArtifactType Property

Returns the ArtifactType object associated with this relationship type.

ArtifactType

Return Value

Returns the ArtifactType object (IArtifactType interface) associated with this relationship type.

Arguments

None

Example

```
Dim theArtifactType As ArtifactType
Dim theRelationship As RelationshipType
theArtifactType = theRelationship.ArtifactType
```

RelationshipType.Cardinality Property

Returns either NARY or UNARY (many or one) for a relationship type.

Cardinality

Return Value

Returns an eRelationshipCardinality type (rsZeroToMany or rsZeroToOne).

Arguments

None

Example

```
Dim ArtifactTypeName As String
Dim theRelationship As RelationshipType
Dim Cardinality As String
  If theRelationship.GetRelatedArtifactType() Is Nothing Then
    ArtifactTypeName = "(unspecified type)"
  Else
    ArtifactTypeName = theRelationship.GetRelatedArtifactType.Name
  End If
  Select Case theRelationship.Cardinality
  Case rsZeroToMany:
    Cardinality = "Zero To Many"
  Case rsZeroToOne:
    Cardinality = "Zero To One"
  End Select
```

RelationshipType.Category Property

Returns a relationship type category.

Category

Return Value

Returns an eRelationshipCategory type (rsPeer, rsDescendant rs, rsChild, or rsAll).

Arguments

None

Example

```
Dim RelCategory As eRelationshipCategory
Dim theRelationship As RelationshipType
RelCategory = theRelationship.Category
Select Case RelCategory
  Case rsAll
    GetRelCategoryName = "All"
  Case rsChild
    GetRelCategoryName = "Child"
  Case rsDescendant
    GetRelCategoryName = "Descendant"
  Case rsPeer
    GetRelCategoryName = "Peer"
End Select
```

RelationshipType.CreationArguments Property

Returns an ArtifactArguments collection suitable for passing to IArtifact.CreateArtifact.

CreationArguments

Return Value

Returns an ArtifactArgumentCollection object (IArtifactArgumentCollection interface).

Arguments

None

Example

```
Dim theRelationship As RelationshipType
Dim theArgCollection As ArtifactArgumentCollection
Dim theArg As ArtifactArgument
Dim ArgumentID As Integer
Set theArgCollection = theRelationship.CreationArguments
For ArgumentID = 0 To theArgCollection.Count - 1
   Set theArg = theArgCollection.Item(ArgumentID)
   theArg.Value = InputBox("Please enter a value for argument " &
theArg.ArgumentName & ".", "RDSI Test", CStr(theArg.DefaultValue))
Next ArgumentID
```

RelationshipType.Key Property

Returns the key of this relationship type. IArtifactType.Key contains the Name property with the spaces and punctuation removed.

Key

Return Value

Returns the Key value of this relationship type.

Arguments

None

Example

```
Dim theRelationship As RelationshipType
Dim NameText As String
   NameText = theRelationship.Name
   If theRelationship.Name <> theRelationship.Key Then
      NameText = NameText + " (" + theRelationship.Key + ")"
   End If
   NameText = JustifyString(NameText, NameWidth, True)
```

RelationshipType.Name Property

Returns the name of this relationship type.

Name

Return Value

Returns the name of this relationship type.

Arguments

None

Example

```
Dim RelationshipID As Integer
Dim theRelationship As RelationshipType
Dim theRelatedType As ArtifactType
Dim NameText As String
Dim TypeText As String
Dim ModText As String

Dim theRelationshipTypes As RelationshipTypeCollection
Dim theRelationship As RelationshipType
Dim NameText As String
For RelationshipID = 0 To theRelationshipTypes.Count - 1
   Set theRelationship = theRelationshipTypes.Item(RelationshipID)
   Set theRelatedType = theRelationship.GetRelatedArtifactType()
   NameText = theRelationship.Name
   If theRelationship.Name <> theRelationship.Key Then
      NameText = NameText + " (" + theRelationship.Key + ")"
   End If
   NameText = JustifyString(NameText, NameWidth, True)
   If theRelatedType Is Nothing Then
      TypeText = JustifyString("(Unspecified)", TypeWidth, True)
   Else
      TypeText = JustifyString(theRelatedType.Name, TypeWidth, True)
   End If
   If theRelationship.CreateAndDeleteAllowed() = True Then
   ModText = "Yes"
   Else
   ModText = "No"
```

```
   End If
PrintResults IndentLevel, NameText + TypeText + ModText
Next RelationshipID
```

RelationshipType.RelationshipID Property

Returns the RelationshipID of the relationship type. This is a unique id for the RelationshipType.

RelationshipID

Return Value

Returns the RelationshipID of the relationship type.

Arguments

None

Example

```
Dim theRelationshipID As Long
Dim theRelationship As RelationshipType
theRelationshipID = theRelationship.RelationshipID
```

RelationshipType Methods

Table 52 lists the RelationshipType methods.

Table 52 IRelationshipType Methods

Method	Description
CreateAndDeleteAllowed	Returns TRUE if artifacts of this relationship type can be created and deleted. Returns FALSE otherwise.
CreateArtifactFilter	Returns an artifact filter object.
GetRelatedArtifactType	Returns the artifact type of the object that this relationship type returns.
IsDynamicType	Returns TRUE if this instance is a dynamic type.

RelationshipType.CreateAndDeleteAllowed Method

Returns TRUE if artifacts of this relationship type can be created and deleted. Returns FALSE otherwise.

CreateAndDeleteAllowed

Return Value

Returns TRUE if artifacts of this relationship type can be created and deleted. Returns FALSE otherwise.

Arguments

None

Example

```
Dim theRelationshipType As RelationshipType
Dim Result As Boolean
Result = theRelationshipType.CanCreateAndDelete
```

RelationshipType.CreateArtifactFilter Method

Returns an ArtifactFilter object.

CreateArtifactFilter (IArtifactFilter Filter)

Return Value

Returns an ArtifactFilter object (IAftifactFilter interface).

Arguments

None

Example

```
Dim theRelationshipType As RelationshipType
Dim theFilter As ArtifactFilter
Set theFilter = theRelationship.CreateArtifactFilter
```

RelationshipType.GetRelatedArtifactType Method

Returns the artifact type of the object that this relationship type returns. Returns NULL if there can be more than one type of Artifacts for this relatonship.

GetRelatedArtifactType

Return Value

Returns an ArtifactType object (IArtifactType interface).

Arguments

None

Example

```
Dim theRelationship As RelationshipType
Dim ArtifactTypeName As String
If theRelationship.GetRelatedArtifactType() Is Nothing Then
   ArtifactTypeName = "(unspecified type)"
Else
   ArtifactTypeName = theRelationship.GetRelatedArtifactType.Name
End If
```

RelationshipType.IsDynamicType Method

Returns TRUE if this instance is a dynamic type. A dynamic type is defined as a type that is specific to a given instance of an artifact. Static types are always available.

IsDynamicType

Return Value

Returns TRUE if this instance is a dynamic type.

Arguments

None

Example

```
Dim theRelationship As RelationshipType
Dim DynamicText As String
If theRelationship.IsDynamicType() = True Then
   DynamicText = "Dynamic"
Else
   DynamicText = "Static"
End If
```

IRelationshipTypeCollection Interface

This is the default interface for the collection of relationship types for an artifact type. This interface encapsulates information about the list of relationship types.

RelationshipTypeCollection Properties

Table 53 lists the RelationshipTypeCollection properties.

Table 53 RelationshipTypeCollection Properties

Property	Description
Count	Returns the count of items in the collection.
Item	Returns a reference to the object at the given index.

RelationshipTypeCollection.Count Property

Returns the count of items in the collection.

Count

Return Value

Returns the count or number of items in the collection.

Arguments

None

Example

```
Dim theRelationshipTypeCollection As RelationshipTypeCollection
Dim theCount26 As Integer
theCount26 = theRelationshipTypeCollection.Count
```

RelationshipTypeCollection.Item Property

Takes a variant index and returns the type of object in the collection. Returns a reference to the object at the given index using its default interface.

Item (VARIANT index)

Return Value

Returns a RelationshipType object (IRelationshipType interface).

Argument

index

A variant index to an item in the collection.

Example

```
Dim theRelationshipType As RelationshipType
Dim theRelationshipTypeCollection As RelationshipTypeCollection
Set theRelationshipType = theRelationshipTypeCollection.Item(1)
```

RelationshipTypeCollection Methods

Table 54 lists the RelationshipTypeCollection methods.

Table 54 IRelationshipTypeCollection Methods

Method	Description
Add	Appends the specified object to the end of the collection.
AddCollection	Appends the contents of the specified collection to this collection.
IsModifiable	Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.
Remove	Removes the object at the specified index from the collection.

RelationshipTypeCollection.Add Method

Appends the specified object to the end of the collection.

Add (IRelationshipType RelationshipType)

Return Value

None

Argument

RelationshipType

A RelationshipType object (IRelationshipType interface).

Example

```
Dim theRelationshipType As RelationshipType
Dim theRelationshipTypeCollection As RelationshipTypeCollection
Set theRelationshipTypeCollection = New RelationshipTypeCollection
theRelationshipTypeCollection.Add theRelationshipType
```

RelationshipTypeCollection.AddCollection Method

Appends the contents of the specified collection to this collection.

AddCollection (IRelationshipTypeCollection Collection)

Return Value

None

Argument

Collection

A RelationshipTypeCollection object (IRelationshipTypeCollection interface).

Example

```
Dim theRelationshipTypeCollection As RelationshipTypeCollection
Dim theRelationshipTypeCollection2 As RelationshipTypeCollection
Set theRelationshipTypeCollection = New RelationshipTypeCollection
theRelationshipTypeCollection.AddCollection
theRelationshipTypeCollection2
```

RelationshipTypeCollection.IsModifiable Method

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise. In general, only user created collections can be modified. Collections returned by RSE objects are typically read-only to the client.

IsModifiable

Return Value

Returns TRUE if the client is able to modify the contents of the collection, FALSE otherwise.

Arguments

None

Example

```
Dim theMod26 As Boolean
Dim theRelationshipTypeCollection As RelationshipTypeCollection
theMod26 = theRelationshipTypeCollection.IsModifiable
```

RelationshipTypeCollection.Remove Method

Removes the object at the specified index from the collection.

Remove (VARIANT index)

Return Value

None

Argument

index

A variant index to an item in the collection.

Example

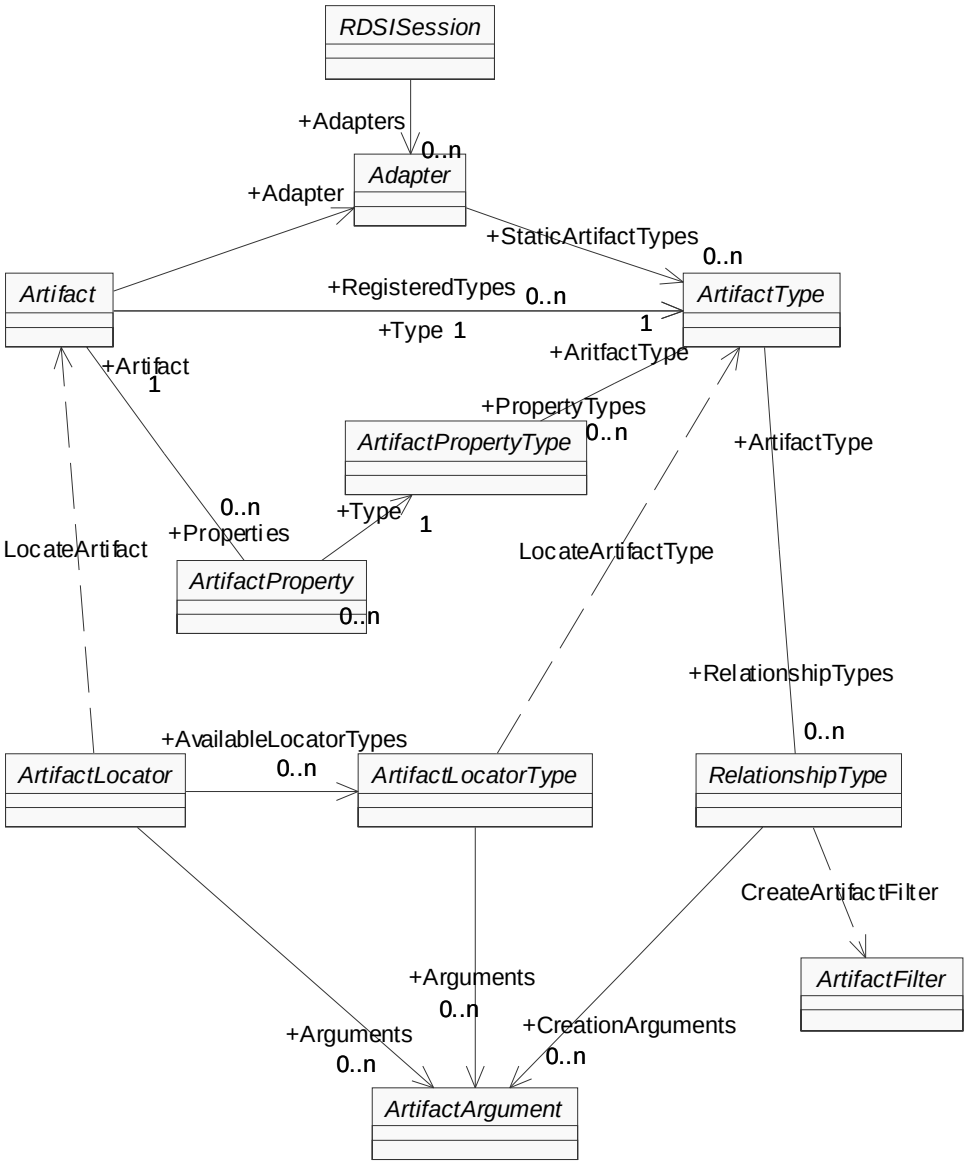
```
Dim theRelationshipTypeCollection As RelationshipTypeCollection
theRelationshipTypeCollection.Remove(1)
```

Class Diagrams

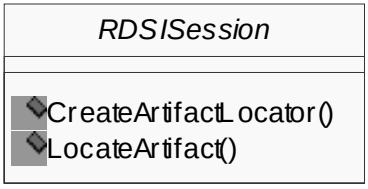
This section includes class diagrams. It includes class diagrams for the following types of interfaces:

- Client interfaces overview
- Session interface
- Adapter interfaces
- Artifact interfaces
- Locator interfaces
- Property interfaces
- Relationship interfaces
- Query interfaces
- Graphics support interfaces

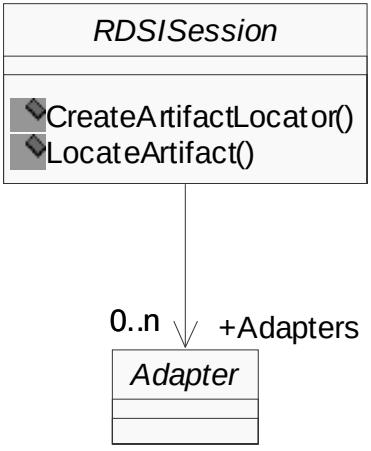
Client Interfaces Overview



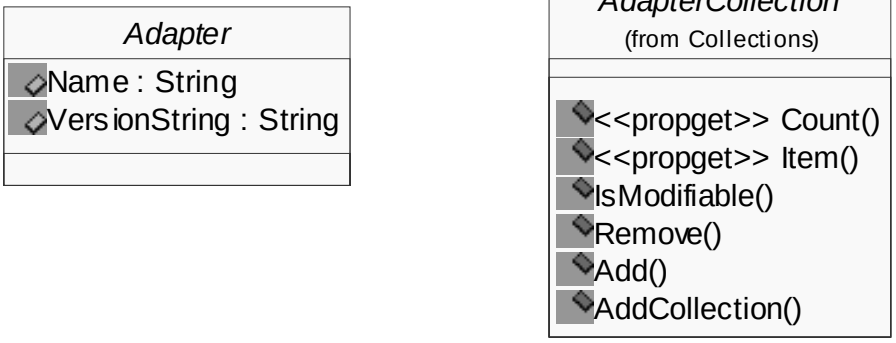
Session Interface



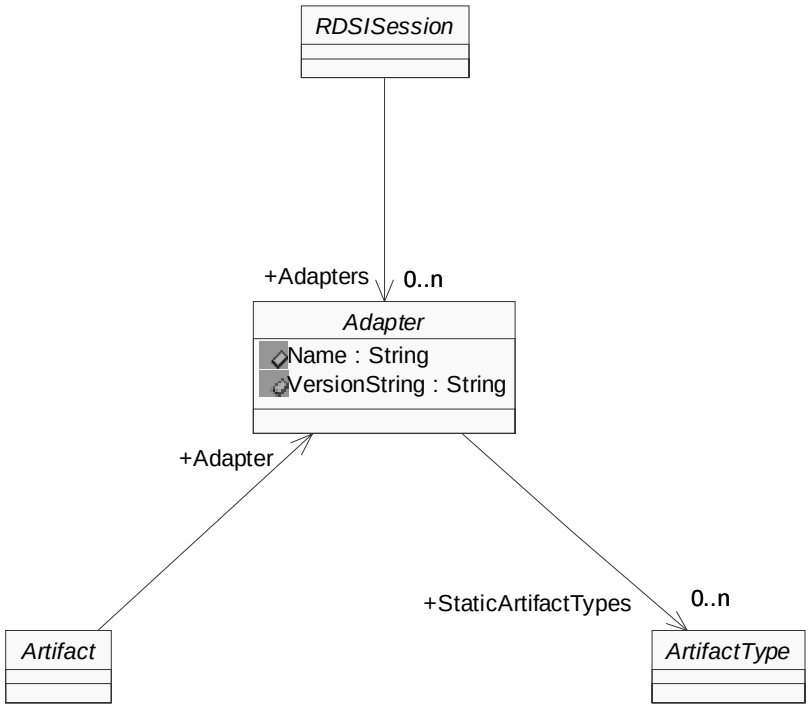
Session Overview



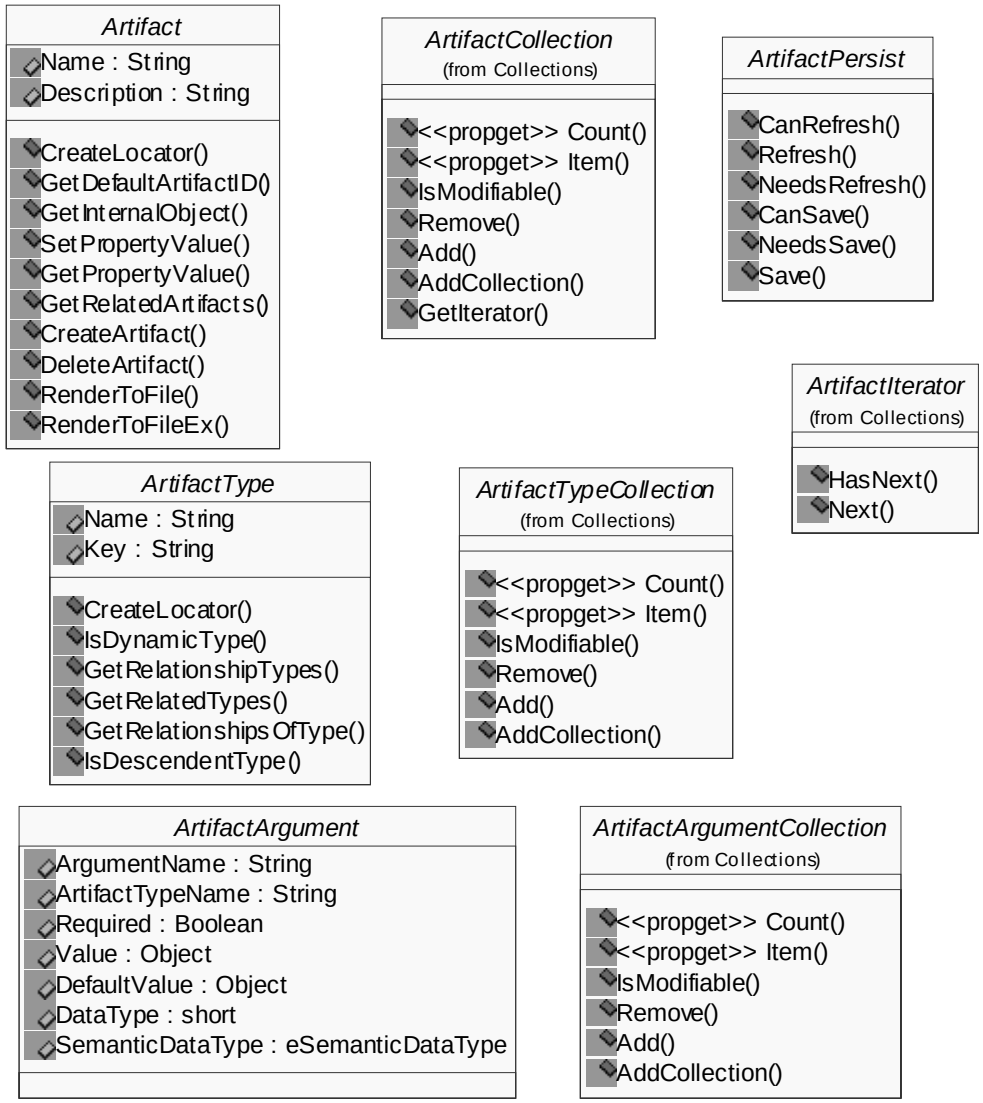
Adapter Interfaces



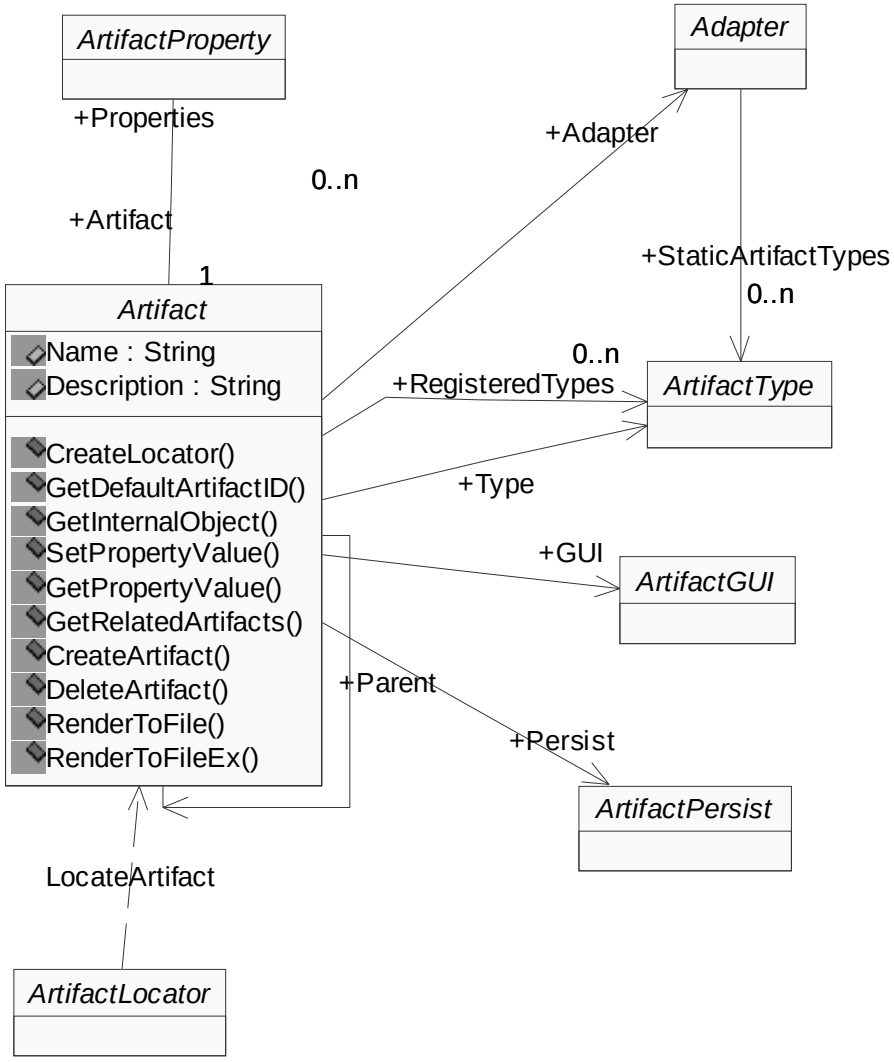
Adapter Overview



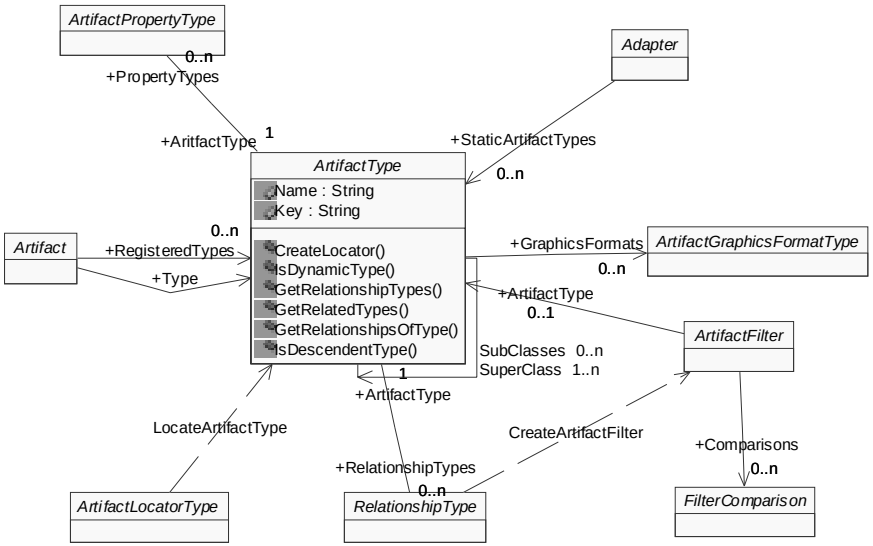
Artifact Interfaces



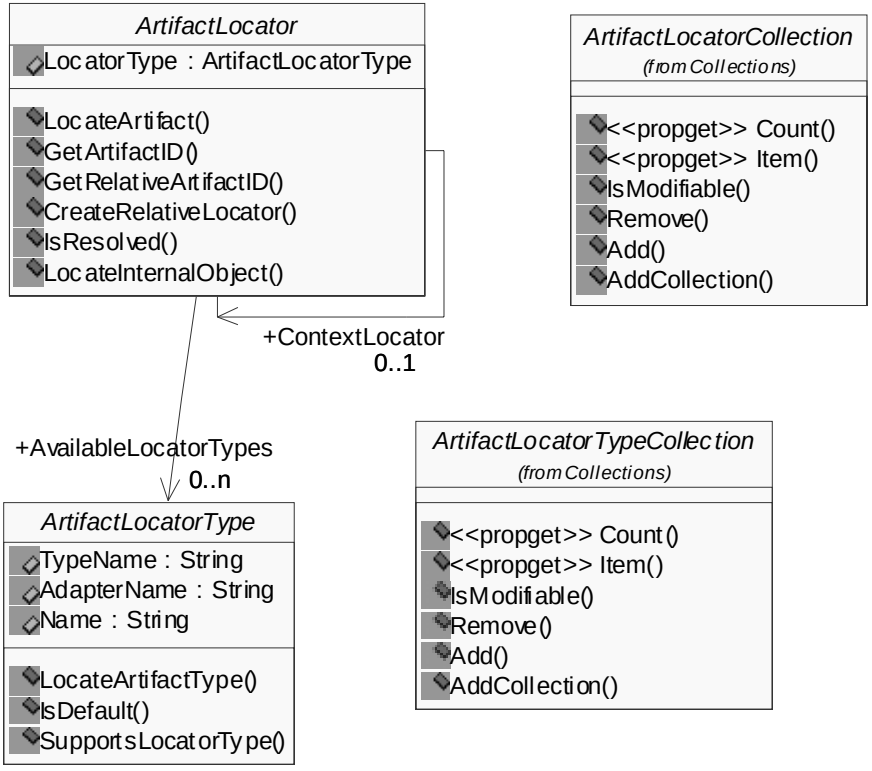
Artifact Overview



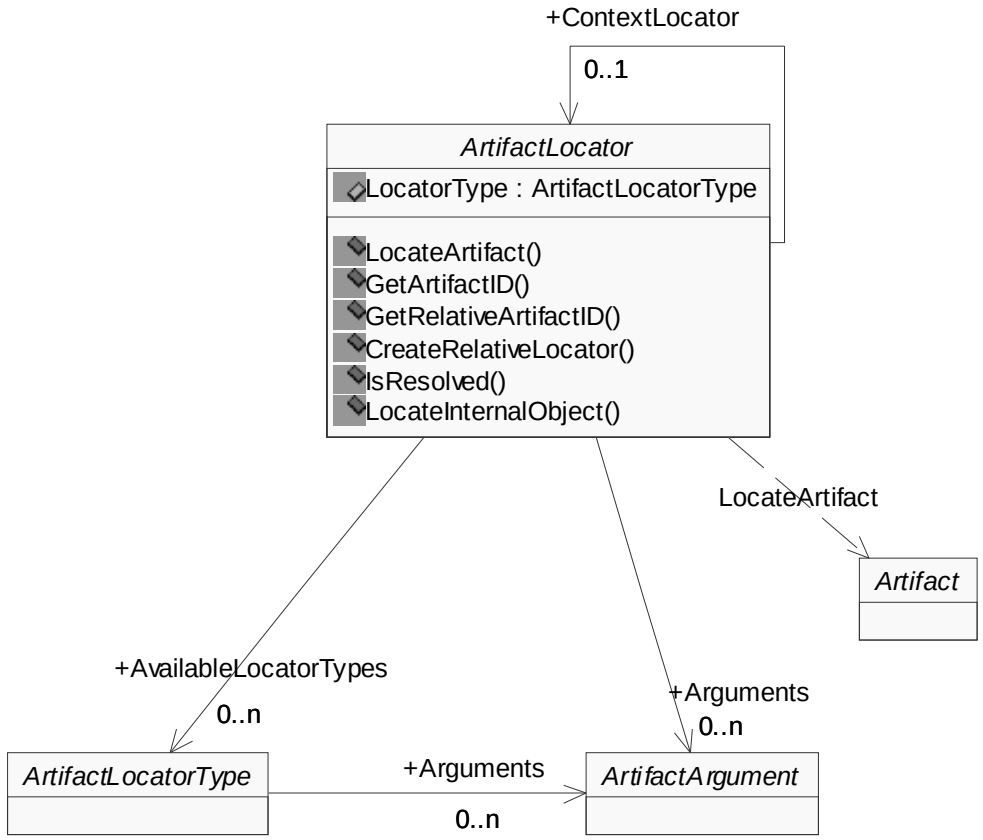
ArtifactType Overview



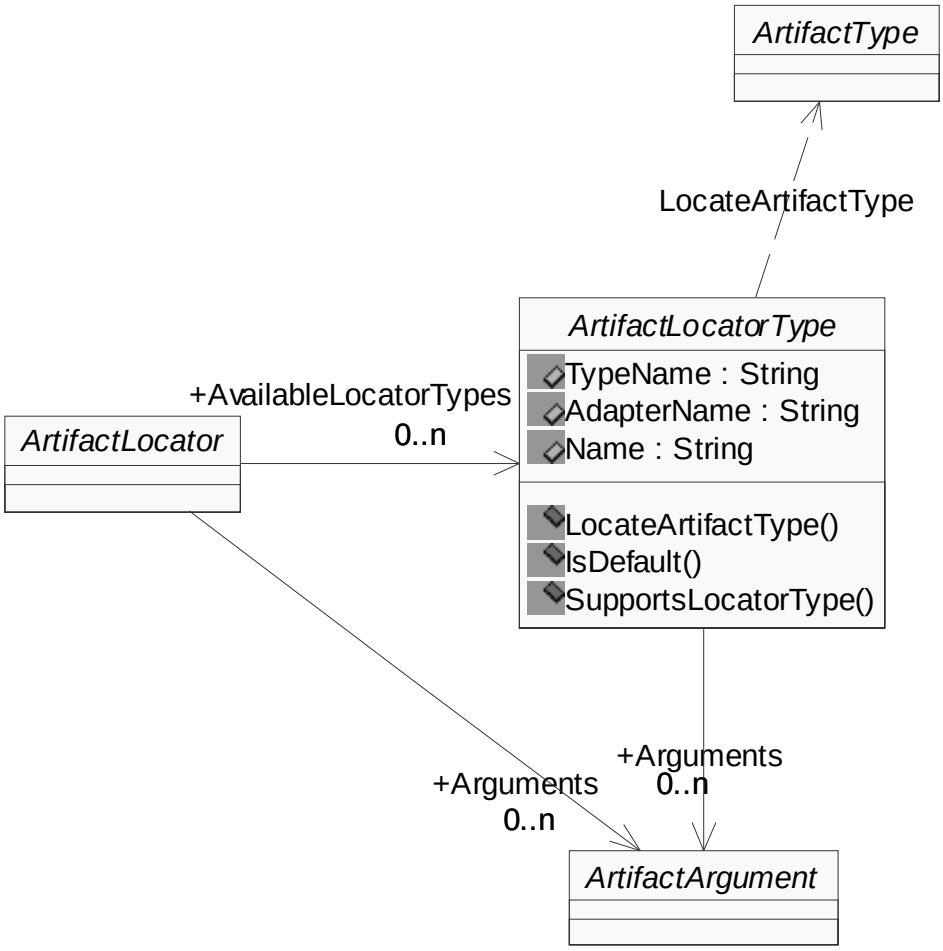
Locator Interfaces



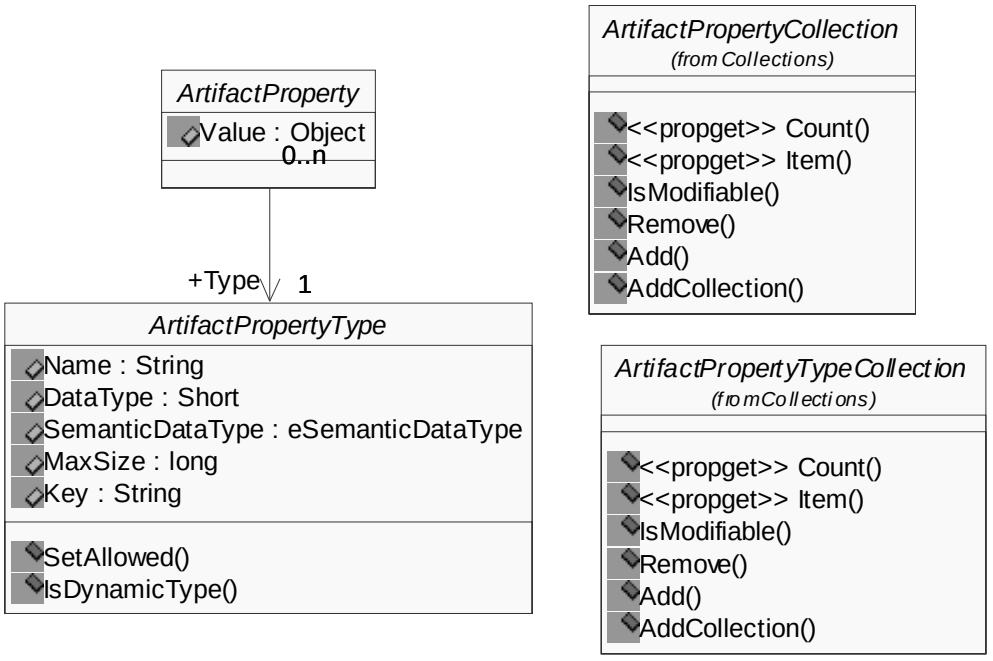
Locator Overview



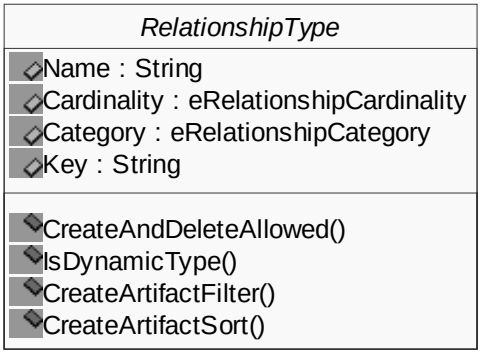
LocatorType Overview

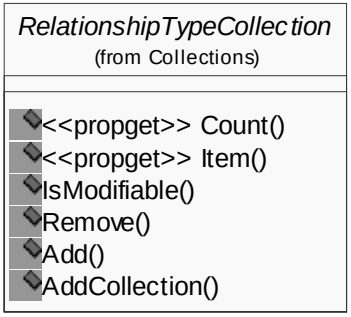


Property Interfaces

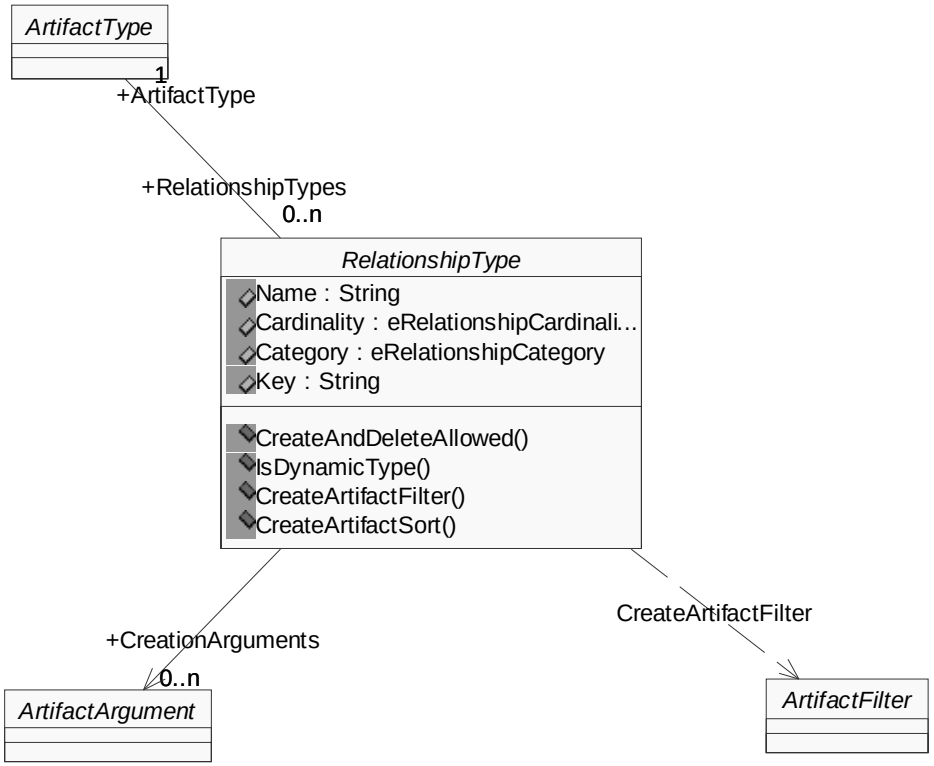


Relationship Interfaces

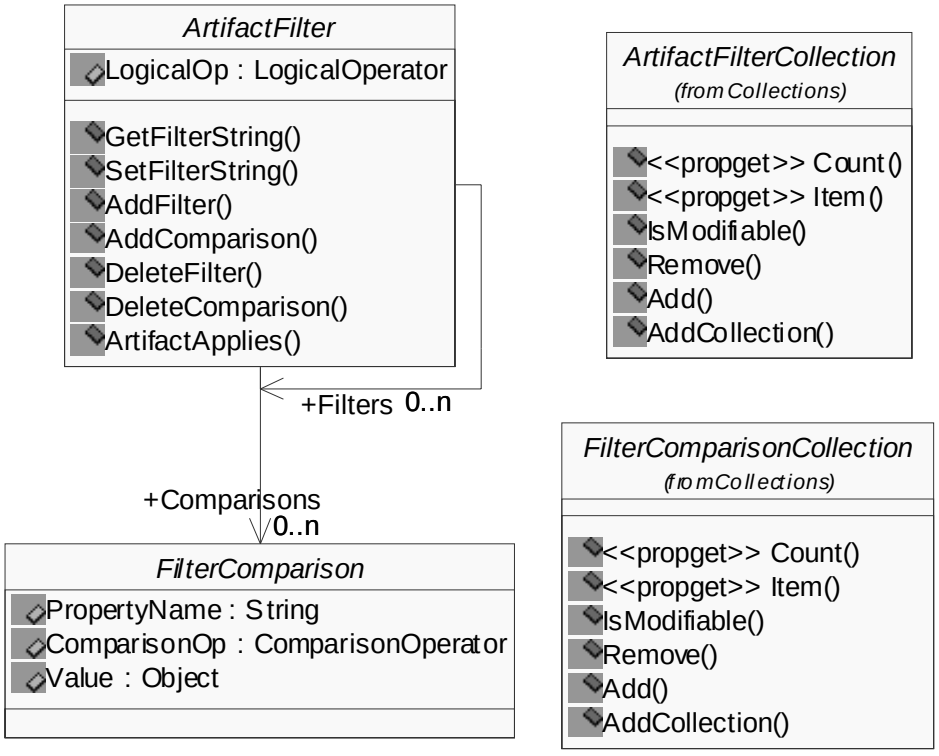




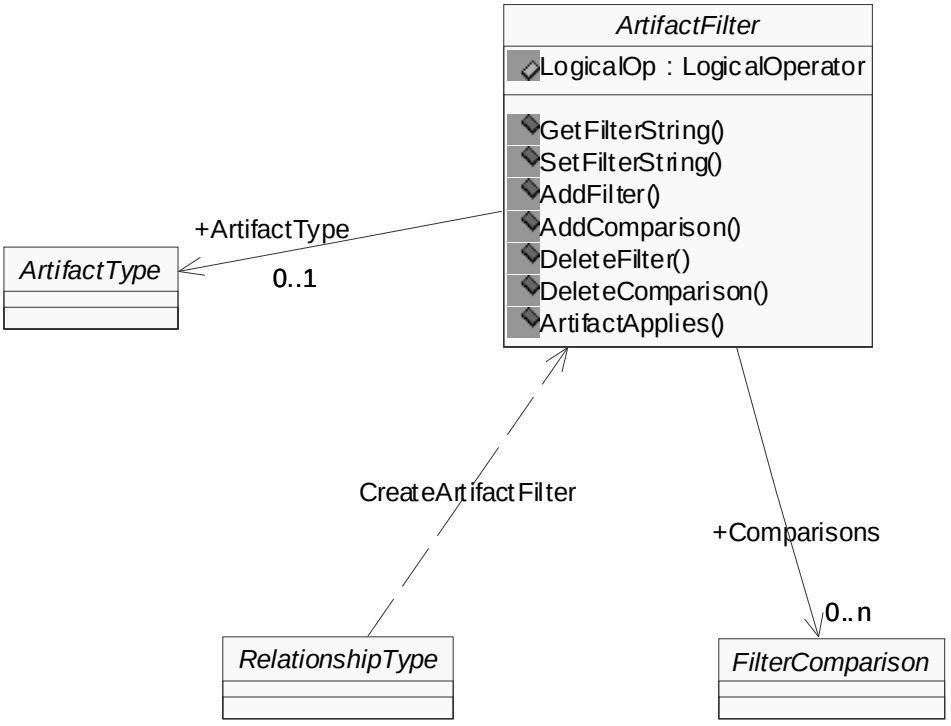
RelationshipType Overview



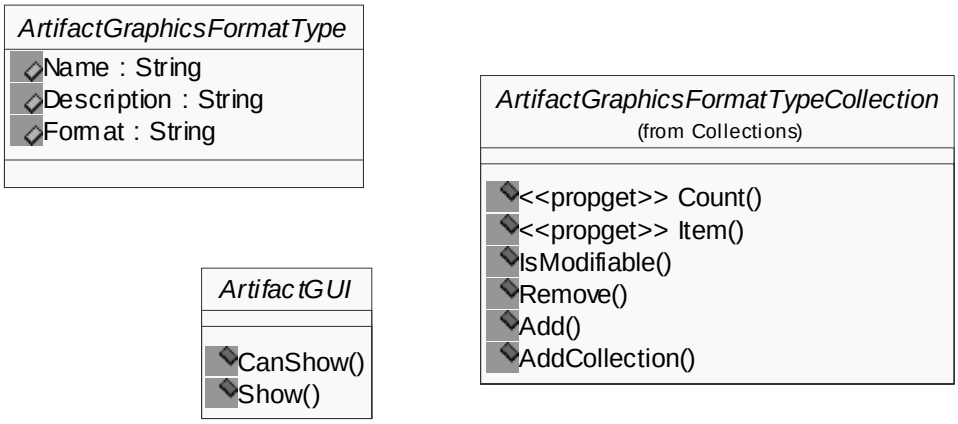
Query Interfaces



ArtifactFilter Overview



Graphics Support Interfaces



Index

A

C

D

F

G

I

L

M

N

P

R

S