API for BIM Exchange
Autodesk Inventor 2011 Subscription Bonus Pack

Contents
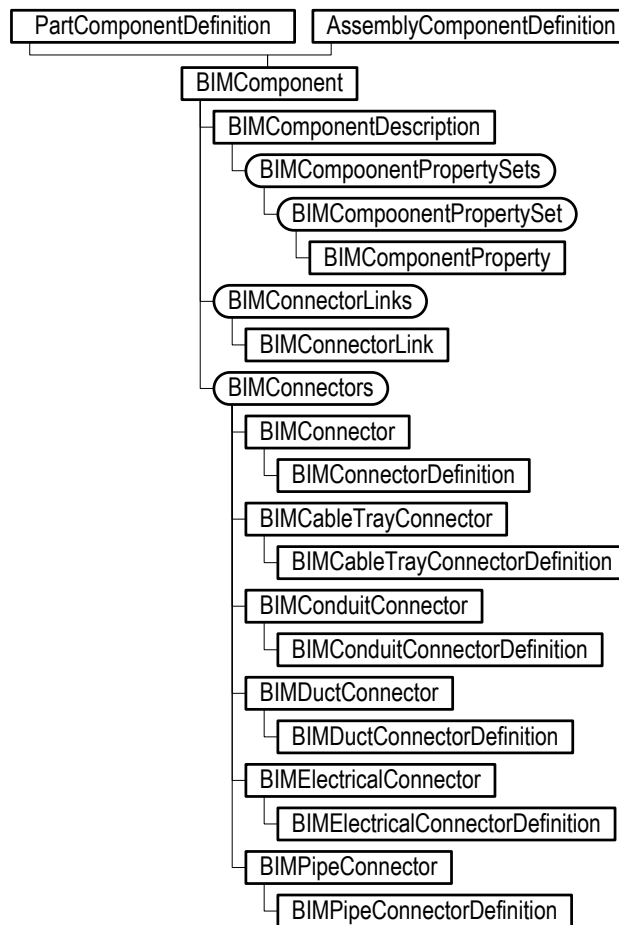
1 Introduction

This document provides a detailed description of the application programming interface (API) for BIM Exchange (previously called AEC Exchange). The following diagram illustrates the object model and how to access it through the PartComponentDefinition and AssemblyComponentDefinition objects.

The BIM Exchange API provides exactly the same capabilities that are provided through the user-interface. If you're attempting to do something using the API and it's failing, try doing the same thing using the user-interface to verify that what you want to do is supported.

The BIM Exchange API is an enhancement to the Inventor API so as long as you are referencing the Inventor Object Library you have access to the BIM Exchange functionality. The type library where the API functionality is defined has been updated with the BIM Exchange functionality as part of the subscription bonus pack. If you are using VBA or VB6 you'll automatically get access to the new functionality. If you are using VC++ you may need to delete your current .tli and .tlh files so that Visual Studio will regenerate them from the new type

library. If you are using a .Net language, there are some additional steps you'll need to do, which are described below.

Inventor's API is exposed through a COM Automation interface. To access this from a .Net language a .Net wrapper must be created, which is referred to as a COM interop assembly. When Inventor is installed it registers a Primary Interop Assembly (PIA) in the Global Assembly Cache (GAC) which means it is making a single interop assembly available to all applications that want to use it. With the BIM Exchange enhancements being added for the subscription bonus pack there were some versioning concerns and <u>a new PIA has not been created for the subscription bonus pack.</u> In order to access the new BIM Exchange functionality you'll need to create your own private interop assembly from Inventor's type library. You do this using the tlbimp utility that's delivered with .Net.

To create the interop assembly for Inventor's type library you use the command line below. This uses the tlbimp utility delivered with .Net.

```
tlbimp "C:\Program Files\Autodesk\Inventor 2011\Bin\Rxinventor.tlb"
/out:My.Inventor.Interop.dll  /namespace:Inventor  /asmversion:15.0.0.0
```

After you've created the interop for the Inventor type library you can reference it into your project using the Add Reference command in Visual Studio. Use the Browse tab to select the interop dll. You'll now have access to the full Inventor API, including the BIM Exchange enhancements. You'll need to deliver this interop dll with your application. Typically this will be in the same location as your application's exe or dll.
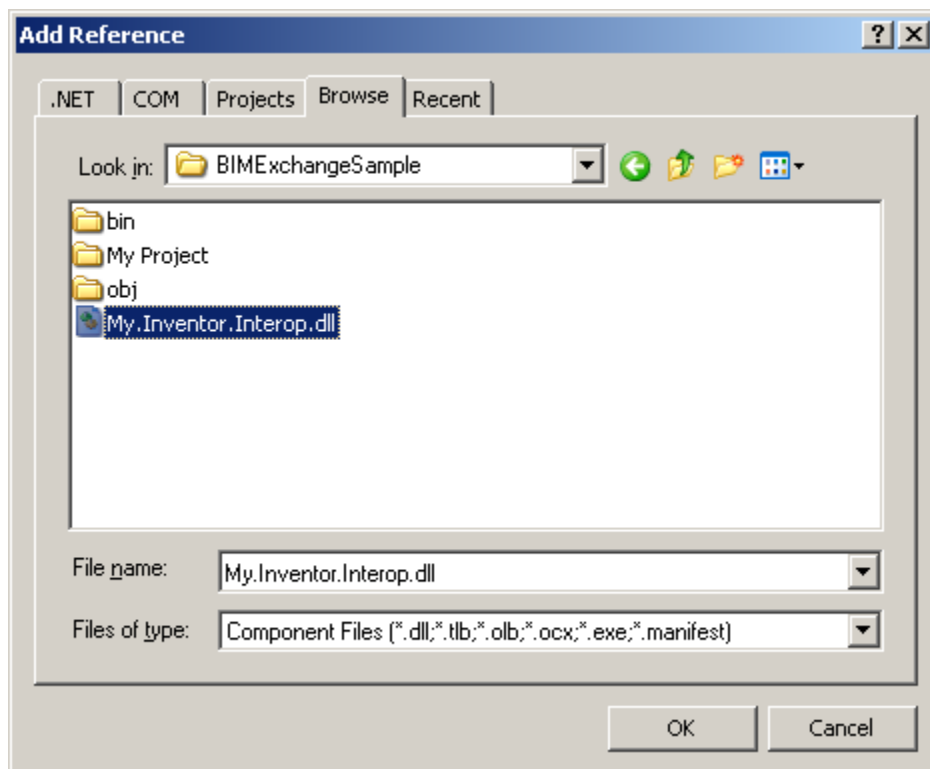
2 PartComponentDefinition Object

BIMComponent As BIMComponent
> Read-only property that returns the BIMComponent object associated with this component definition.

3 AssemblyComponentDefinition Object

BIMComponent As BIMComponent
> Read-only property that returns the BIMComponent object associated with this component definition.

4 BIMExchangeServer Object

This is the object returned by the BIM Exchange add-in through the Automation property.

GetBIMComponent (*ComponentDefinition* As ComponentDefinition) As BIMComponent
> Method that returns the BIMComponent associated with the provided Inventor part or assembly component definition.

Parameter	Type	Tags	Description
ComponentDefinition	ComponentDefinition	In	Input PartComponentDefinition or AssemblyComponentDefinition object that specifies which Inventor part or assembly you want to get the associated BIMComponent for.

5 BIMComponent Object

Object that provides access to all of the BIM related information associated with an Inventor document.

Application As Object
> Read-only property that returns the root Application object.

ComponentDefinition As ComponentDefinition
> Read-only property that returns the Inventor PartComponentDefinition or AssemblyComponentDefinition that this BIMComponent object is associated with.

ComponentDescription As BIMComponentDescription
> Read-only property that returns the BIMComponentDescription object associated this document.

ConnectorLinks As BIMConnectorLinks
> Read-only property that returns the collection of connector links for this document. Through the returned object you can access all existing links between connectors and create new links between connectors.

Connectors As BIMConnectors
> Read-only property that returns the collection of connectors for this document. Through the returned object you can access all existing connectors and create new connectors.

ExportBuildingComponent (*FullFilename* As String)
> Method that exports the BIM component as an adsk file.

Parameter	Type	Tags	Description
FullFilename	String	In	Input String that defines the full filename to write the building component to. The filename should have an adsk extension.

Type As ObjectTypeEnum
Read-only property returning kBIMComponentObject indicating this object's type.

6 BIMComponentDescription Object

Object that provides access to component description of the document.

Application As Object
Read-only property that returns the root Application object.

ComponentPropertySets As BIMComponentPropertySets
Read-only property that returns the collection object containing the property sets for a BIM component.

ComponentType As String
Read-write property that gets and sets the component type. A component type is identified by a category ID. The list of available components is defined in the file:
[Design Data]\BIM Exchange\Support\AuthoringTemplates\Categories.xml

An example of a valid component type is "23.30.10.11.24.11", which specifies the "Structural Doors Sections" category.

ModelProperties As ObjectCollection
Read-write property that gets and sets the standard Inventor iProperties that are exported with this component.

Orientation As UserCoordinateSystem
Read-write property that defines which coordinate system to use when exporting the component. By default the value of this property is Nothing, which indicates that the model origin is to be used.

SetCustomThumbnail(FullFilename As String)
Method that lets you set the custom thumbnail by providing the filename of an image file. The file should be a bmp, jpg, or png format and should be 256x256 pixels.

Thumbnail As IPictureDisp
Read-write property that gets and sets the thumbnail for this component. The bitmap should be file should be a bmp, jpg, or png format and should be 256x256 pixels. Setting this property to Nothing will result in an automatic thumbnail being generated.

This property can only be used when your application is running in the same process as the Inventor application. If you're running out of process then you can use the SetCustomThumbnail property to set the thumbnail using a file on disk.

Type As ObjectTypeEnum
Read-only property returning kBIMComponentDescription Object indicating this object's type.

7 BIMComponentPropertySets Object

Collection object that provides access to all of the BIM property sets associated with this component.

Application As Object

Read-only property that returns the root Application object.

Count As Long
Read-only property that returns the total number of BIM component property sets this BIM document.

Item (*Index* As Variant) As BIMComponentPropertySet
Read-only property that returns the specified BIMComponentPropertySet object from the collection. This collection is empty until the component type is defined using the ComponentType property of the BIMComponentDescription object.

Parameter	Type	Tags	Description
Index	Variant	In	Input Variant value that specifies the BIM component property set to return. This is the index of the item in the collection where the first item is 1. It can also be the name or internal name of the property set. If an out of range value or an unknown name is provided an error will occur.

Type As ObjectTypeEnum
Read-only property returning kBIMComponentPropertySets indicating this object's type.

8 BIMComponentPropertySet Object

Collection object that provides access to all of the BIM properties associated with this BIM property set.

Application As Object
Read-only property that returns the root Application object.

Count As Long
Read-only property that returns the total number of BIM component property sets this BIM document.

InternalName As String
Read-only property that gets the internal name of this property set. This name is consistent and can be used as a reliable index for this property set.

Item (*Index* As Variant) As BIMComponentProperty
Read-only property that returns the specified BIMComponentProperty object from the collection.

Parameter	Type	Tags	Description
Index	Variant	In	Input Variant value that specifies the BIM component property to return. This is the index of the item in the collection where the first item is 1. It can also be the name or internal name of the property. If an out of range value or an unknown name is provided an error will occur.

Name As String
Read-only property that gets the visible name of this property set. This is the name shown to the end-user in the Component Properties list in the Export Building Components dialog. This name is localized and can change for different languages.

Parent As **BIMComponent**
Read-only property that returns the parent BIMComponent object of this object.

Type As ObjectTypeEnum
Read-only property returning kBIMComponentPropertySet indicating this object's type.

9 BIMComponentProperty Object

Object that represents a single BIM property. These are the properties shown in the Component Properties area of the Export Building Components dialog.

Application As Object
> Read-only property that returns the root Application object.

InternalName As String
> Read-only property that gets the internal name of this component property. This name is consistent and can be used as a reliable index for this property.

Name As String
> Read-only property that gets the visible name of this property. This is the name shown to the end-user in the Component Properties list in the Export Building Components dialog. This name is localized and can change for different languages.

Parent As **BIMComponentPropertySet**
> Read-only property that returns the parent BIMComponentPropertySet object of this object.

Type As ObjectTypeEnum
> Read-only property returning kBIMComponentProperty indicating this object's type.

Value As String
> Read-write property that gets the value of this BIM property.

10 BIMConnectors Object

Collection object that provides access to all of the BIM connectors within a document. Methods on this collection are also used to create new BIM connectors.

Application As Object
> Read-only property that returns the root Application object.

Add (*Definition* As BIMConnectorDefinition, Optional *Name* As String = "") As BIMConnector
> Method that creates a new BIMConnector. The type of connector definition supplied will determine the type of connector created. The new BIMConnector is returned.

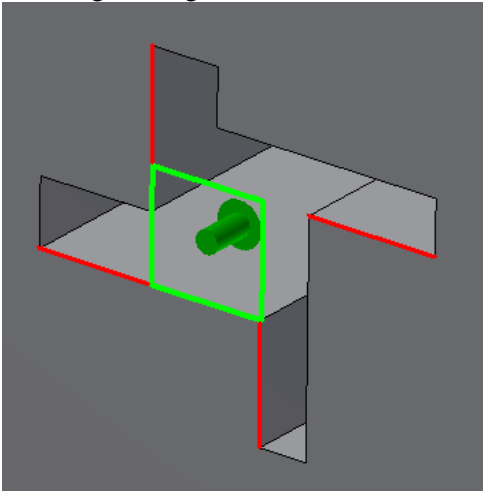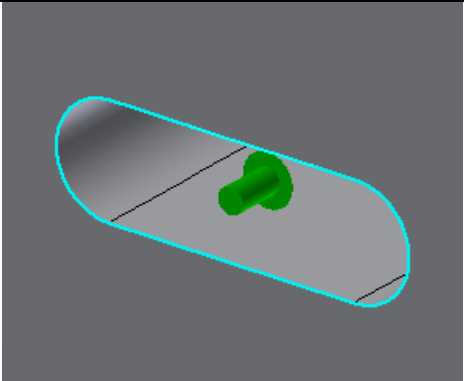| Parameter	Type	Tags	Description
Definition	BIMConnectorDefinition	In	Input definition object that defines all of the required inputs to create a new connector.
Name	String	Optional In	Defines the name of the new connector. It must be unique with respect to all existing connectors. If not provided, or an empty string is supplied then BIM Exchange will generate a default name.

Count As Long
> Read-only property that returns the total number of all types of connectors in this BIMDocument.

CreateCableTrayConnectorDefinition (*Geometry* As ObjectCollection, *ConnectorShape* As BIMConnectorShapeEnum) As BIMCableTrayConnectorDefinition

---

Method that creates a new cable tray connector definition. The created definition object defines the inputs to create a cable tray connector and is used as input to the Add method of the BIMConnectors object to create a new connector.

The created definition defaults to a rectangular shape, the height and width are defined by the input geometry, and the connection type is electrically bonded. You can change any of these settings by using the methods and properties on the returned BIMCableTrayConnectorDefinition.

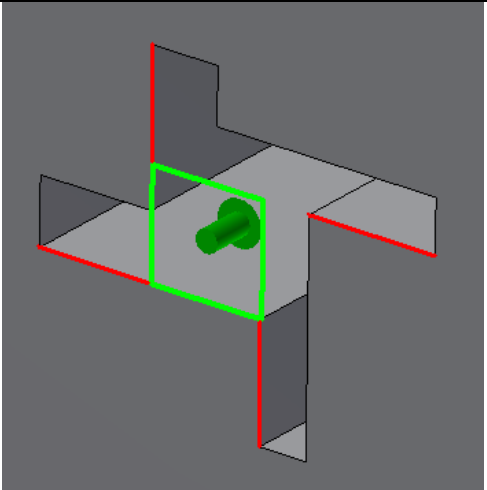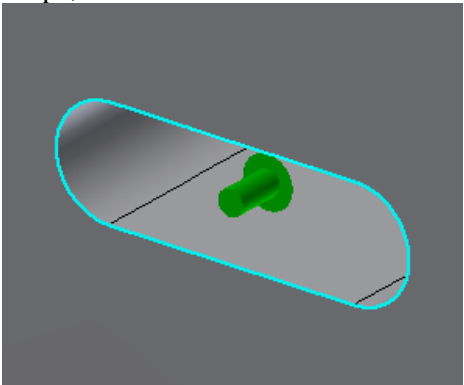| Parameter	Type	Tags	Description
Geometry	ObjectCollection	In	Input object collection that contains the geometry that defines the shape of the connector. When the specified shape is rectangular, valid input includes a single rectangular face or four linear edges on a planar face that define a rectangle. These edges do not need to connect but two of them need to be parallel to each other and perpendicular to the other two edge. For example, the picture below illustrates four valid lines and the resulting rectangle. When the specified shape is undefined, valid input includes the input described above for a rectangular shape but also allows a circular planar face, a circular edge (it can be an arc), a planar face that has a slot shape, or four edges that define a slot or oval shape, as illustrated below.

ConnectorShape	BIMConnectorShapeEnum	In	Input value that specifies the shape of the connector. This setting affects how the geometry is evaluated to determine the shape of the connector. The following inputs are valid for a cable tray connector: kRectangularShapeConnector or kUndefinedShapeConnector

CreateConduitConnectorDefinition (*Geometry* As ObjectCollection, *ConnectorShape* As BIMConnectorShapeEnum) As BIMConduitConnectorDefinition

Method that creates a new conduit connector definition. The created definition object defines the inputs to create a conduit connector and is used as input to the Add method of the BIMConnectors object to create a new conduit connector.

The created definition defaults to a circular shape and the diameter is defined by the input geometry. You can change any of these settings by using the methods and properties on the returned BIMConduitConnectorDefinition.

Parameter	Type	Tags	Description
Geometry	ObjectCollection	In	Input object collection that contains the geometry that defines the shape of the connector. When the specified shape is circular, valid input includes a single circular face or a circular edge (it can be an arc). When the specified shape is undefined, valid input includes the input described above for a rectangular shape but also allows a rectangular planar face, four edges that define a rectangle as illustrated below,

a planar face that has a slot shape, or four edges that define a slot or oval shape, as illustrated below.

Parameter	Type	Tags	Description
ConnectorShape	BIMConnectorShapeEnum	In	Input value that specifies the shape of the connector. This setting affects how the geometry is evaluated to determine the shape of the connector. The following inputs are valid for a conduit connector: kCircularShapeConnector or kUndefinedShapeConnector

CreateDuctConnectorDefinition (*Geometry* As ObjectCollection, *ConnectorShape* As BIMConnectorShapeEnum) As BIMDuctConnectorDefinition

Method that creates a new duct connector definition. The created definition object defines the inputs to create a duct connector and is used as input to the Add method of the BIMConnectors object to create a new connector.

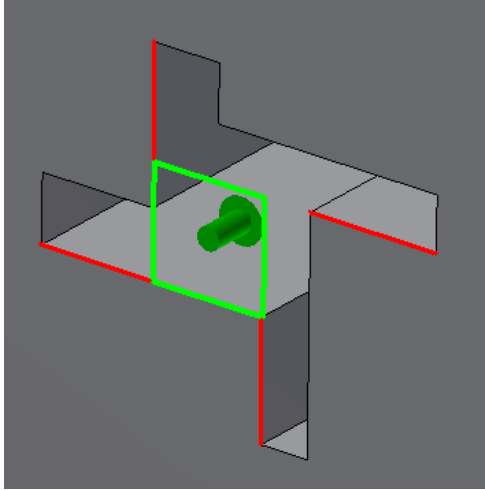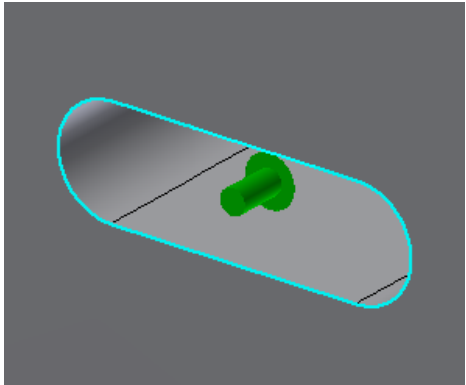| Parameter	Type	Tags	Description
Geometry	ObjectCollection	In	Input object collection that contains the geometry that defines the shape of the connector. When the specified shape is rectangular, valid input includes a single rectangular face or four linear edges on

API for BIM Exchange

a planar face that define a rectangle. These edges do not need to connect but two of them need to be parallel to each other and perpendicular to the other two edge. For example, the picture below illustrates four valid lines and the resulting rectangle.

When the specified shape is oval, valid input includes a face that defines an oval or four edges that define an oval as illustrated below.

When the specified shape is circular, valid input includes a single circular face or a circular edge (it can be an arc).

When the specified shape is undefined, valid input includes any of the above.

ConnectorShape	BIMConnectorShapeEnum	In	Input value that specifies the shape of the connector. This setting affects how the geometry is evaluated to determine the shape of the connector. The

			following inputs are valid for a duct connector: kRectangularShapeConnector, kCircularShapeConnector, kOvalShapeConnector or kUndefinedShapeConnector

CreateElectricalConnectorDefinition (*Geometry* As ObjectCollection) As
BIMElectricalConnectorDefinition

Method that creates a new electrical connector definition. The created definition object defines the inputs to create a electrical connector and is used as input to the Add method of the BIMConnectors object to create a new connector.
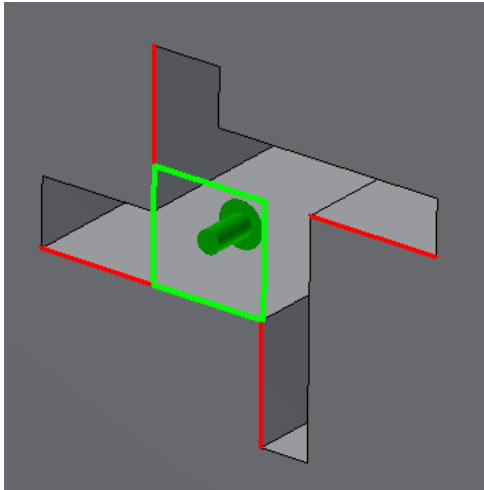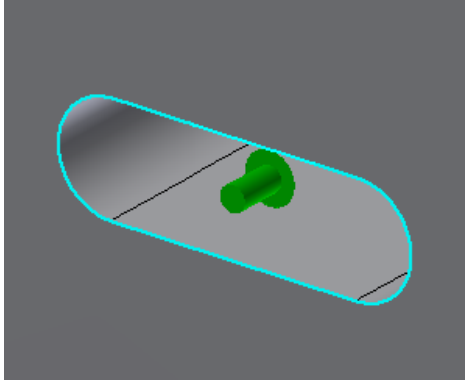
Parameter	Type	Tags	Description
Geometry	ObjectCollection	In	Input object collection that contains the geometry that defines the shape of the connector. Valid input includes: • Single circular face. • Circular edge (it can be an arc) • Rectangular planar face. • Four edges that define a rectangle as illustrated below. • Planar face that has a slot shape. • Four edges that define a slot or oval shape, as illustrated below.
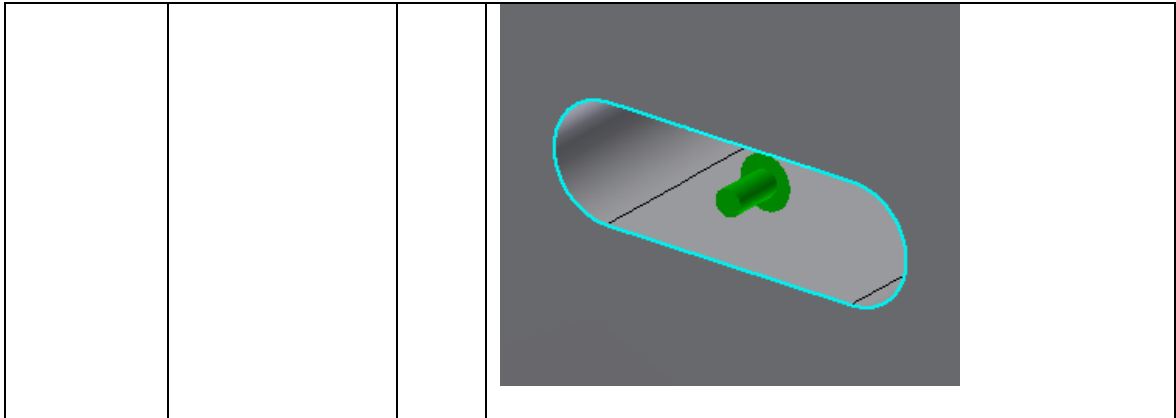
CreatePipeConnectorDefinition (*Geometry* As ObjectCollection, *ConnectorShape* As BIMConnectorShapeEnum) As BIMPipeConnectorDefinition

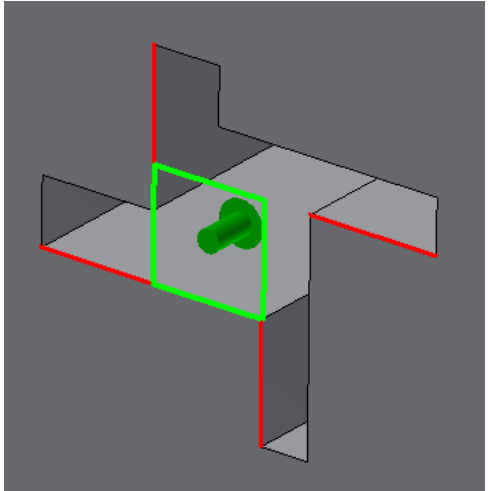> Method that creates a new pipe connector definition. The created definition object defines the inputs to create a pipe connector and is used as input to the Add method of the BIMConnectors object to create a new connector.

Parameter	Type	Tags	Description
Geometry	ObjectCollection	In	Input object collection that contains the geometry that defines the shape of the connector. When the specified shape is circular, valid input includes a single circular face or a circular edge (it can be an arc). When the specified shape is undefined, valid input includes the input described above for a rectangular shape but also allows a rectangular planar face, four edges that define a rectangle as illustrated below, a planar face that has a slot shape, or four edges that define a slot or oval

			shape, as illustrated below.
ConnectorShape	BIMConnectorShapeEnum	In	Input value that specifies the shape of the connector. This setting affects how the geometry is evaluated to determine the shape of the connector. The following inputs are valid for a pipe connector: kCircularShapeConnector or kUndefinedShapeConnector

Item (*Index* As Variant) As BIMConnector
 Read-only property that returns the specified BIMConnector object from the collection.

Parameter	Type	Tags	Description
Index	Variant	In	Input Variant value that specifies the BIM connector to return. This is the index of the item in the collection where the first item is 1. It can also be the name of the connector. If an out of range value or an unknown name is provided an error will occur.

Type As ObjectTypeEnum
 Read-only property returning kBIMConnectors indicating this object's type.

11 BIMConnector Object

Base class object for all of the various BIM connector objects.

Application As Object
 Read-only property that returns the root Application object.

Definition As BIMConnectorDefinition
 Read-only property that gets the definition object associated with this connector.

DefinitionType As BIMConnectorDefinitionTypeEnum
 Read-only property that returns the type of definition associated with this connector. This property lets you determine what type of connector this object represents.

Delete
 Method that deletes the connector.

GetReferenceKey (*ReferenceKey()* As Byte, Optional *KeyContext* As Long)
 Method that generates and returns the reference key for this entity. A reference key is an array of bytes that can be used as a persistent reference for an entity. To obtain the entity at a later time using the reference key you use the BindKeyToObject method of the object. The

ReferenceKeyManager object is obtained using the ReferenceKeyManager property of the Document object.

Parameter	Type	Tags	Description
ReferenceKey	Byte()	Out	Output array of Bytes that contains the reference key.
KeyContext	Long	In	Input Long that specifies the key context. The key context is used for B-Rep entities (SurfaceBody, FaceShell, Face, Edge, EdgeUse and Vertex objects) but is ignored for other entity types, including BIMConnector objects.

Name As String
Read-write property that gets and sets the displayed name of the connector. This is the name that is visible in the browser and is editable by the end-user.

Parent As BimComponent
Read-only property that returns the parent BimComponent object.

Suppressed As Boolean
Read-write property that defines whether the connector is suppressed or not. A value of True indicates the connector is suppressed.

Type As ObjectTypeEnum
Read-only property returning kBIMConnector indicating this object's type.

12 BIMConnectorDefinition Object

Base class object for all of the various BIM connector definition objects.

Application As Object
Read-only property that returns the root Application object.

ConnectorShape As BIMConnectorShapeEnum
Read-only property that specifies the shape of the connector. To change the shape, use the SetShape method.

This setting affects how the geometry is evaluated to determine the shape of the connector. The following inputs are valid for the various types of connectors:

- Cable tray connectors: kRectangularShapeConnector or kUndefinedShapeConnector
- Conduit connectors: kCircularShapeConnector or kUndefinedShapeConnector
- Duct connectors: kRectangularShapeConnector, kCircularShapeConnector, kOvalShapeConnector or kUndefinedShapeConnector
- Electrical connectors: kUndefinedShapeConnector
- Pipe connectors: kCircularShapeConnector or kUndefinedShapeConnector

Direction As UnitVector
Read-only property that indicates the direction of the connection. This property will return Nothing in the case where a valid set of referenced geometries have not yet been defined.

Geometry As ObjectCollection
Read-only property that gets the geometry that defines the shape of the connection. The returned collection is independent of the connector and any changes made to the contents of the collection

will not affect the connector. To change the geometry or the shape of the connector use the SetShape method.

Parent As BIMConnector
Property that returns the parent BIMConnector that this definition is associated with. If the definition was created using one of the Create methods this property will return Nothing since the definition isn't associated with a connector yet.
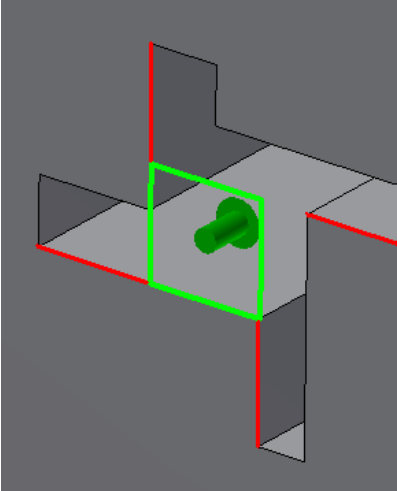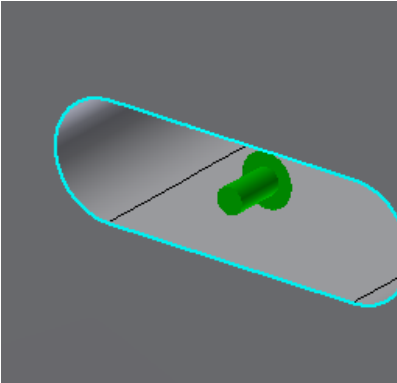
ReverseDirection
Method that will reverse the direction of the connection.

SetShape (*Geometry* As ObjectCollection, Optional *ConnectorShape* As BIMConnectorShapeEnum = kUndefinedShapeConnector) As BIMComponentProperty
Read-only property that returns the specified BIMComponentProperty object from the collection.

Parameter	Type	Tags	Description
Geometry	ObjectCollection	In	Input ObjectCollection that specifies the set of geometry used to define the connector position and shape.
ConnectorShape	BIMConnectorShapeEnum	Optional In	Input value that indicates the desired shape of the connector and how the geometry is to be evaluated. Valid values for the various types of connectors is shown below. - Cable tray connectors: kRectangularShapeConnector or kUndefinedShapeConnector - Conduit connectors: kCircularShapeConnector or kUndefinedShapeConnector - Duct connectors: kRectangularShapeConnector, kCircularShapeConnector, kOvalShapeConnector or kUndefinedShapeConnector - Electrical connectors: kUndefinedShapeConnector - Pipe connectors: kCircularShapeConnector or kUndefinedShapeConnector The input geometry for the various shapes can be the following:

				• Circular shape
				○ Single circular face.
				○ Circular edge (it can be an arc)
				• Rectangular shape
				○ Rectangular planar face.
				○ Four edges that define a rectangle as illustrated below.
				• Slot shape
				○ Planar face that has a slot shape.
				○ Four edges that define a slot or oval shape, as illustrated below.

Type As ObjectTypeEnum

Read-only property returning kBIMConnectorDefinition indicating this object's type.

13 BIMCableTrayConnectorDefinition Object

A programming object that represents the various settings that can be defined for a cable tray connector. This object doesn't represent the actual connector but only represents the settings and is used in the creation, query, and edit of a connector.

Application As Object
> Read-only property that returns the root Application object.

ConnectionType As BIMCableTrayConnectionTypeEnum
> Read-write property that specifies the connection type for this cable tray connector.

Height As Variant
> Read-only property that provides access to the connector height. When the BIMCableTrayConnectorDefinition object has been created using the CreateCableTrayConnectorDefinition method, this property returns a Double indicating the height, (in centimeters), of the connector as defined by the input geometry. After the definition object has been used to create a connector, this property returns a parameter that defines the height of the connector.
>
> To change the height of an existing connector you can either edit the geometry that's controlling the height or set the override height using the HeightOverride property.

HeightOverride As Variant
> Read-write property that gets and sets the height override for this connector. If the value returned by this property is an empty string (""), this indicates that there is no override for the height. Setting this property to an empty string will remove an override. If there is an override this property will return the expression of the parameter controlling the height.
>
> The value can be set using either a double or a string. If a double is input, the units are centimeters. If a string is input, the units can be specified as part of the string or it will default to the current length units of the document. When using a string you can also define equations and reference existing parameters. For example, the string "d5 / 4" is valid, assuming there is a parameter named d5.

Type As ObjectTypeEnum
> Read-only property returning BIMCableTrayConnectorDefinitionObject indicating this object's type.

Width As Variant
> Read-only property that provides access to the connector width. When the BIMCableTrayConnectorDefinition object has been created using the CreateCableTrayConnectorDefinition method, this property returns a Double indicating the width, (in centimeters), of the connector as defined by the input geometry. After the definition object has been used to create a connector, this property returns a parameter that defines the width of the connector.
>
> To change the width of an existing connector you can either edit the geometry that's controlling the width or set the override width using the WidthOverride property.

WidthOverride As Variant
> Read-write property that gets and sets the width override for this connector. If the value returned by this property is an empty string (""), this indicates that there is no override for the width. Setting this property to an empty string will remove an override. If there is an override this property will return the expression of the parameter controlling the width.

The value can be set using either a double or a string. If a double is input, the units are centimeters. If a string is input, the units can be specified as part of the string or it will default to the current length units of the document. When using a string you can also define equations and reference existing parameters. For example, the string "d5 / 4" is valid, assuming there is a parameter named d5.

14 BIMConduitConnectorDefinition Object

A programming object that represents the various settings that can be defined for a conduit connector. This object doesn't represent the actual connector but only represents the settings and is used in the creation, query, and edit of a connector.

Application As Object
> Read-only property that returns the root Application object.

ConnectionType As BIMConduitConnectionTypeEnum
> Read-write property that specifies the connection type for this conduit connector.

Diameter As Variant
> Read-only property that provides access to the connector diameter. When the BIMConduitConnectorDefinition object has been created using the CreateConduitConnectorDefinition method, this property returns a Double indicating the diameter, (in centimeters), of the connector as defined by the input geometry. After the definition object has been used to create a connector, this property returns a parameter that defines the diameter of the connector.
>
> To change the diameter of an existing connector you can either edit the geometry that's controlling the diameter or set the override diameter using the DiameterOverride property.

DiameterOverride As Variant
> Read-write property that gets and sets the diameter override for this connector. If the value returned by this property is an empty string (""), this indicates that there is no override for the diameter. Setting this property to an empty string will remove an override. If there is an override this property will return the expression of the parameter controlling the diameter.
>
> The value can be set using either a double or a string. If a double is input, the units are centimeters. If a string is input, the units can be specified as part of the string or it will default to the current length units of the document. When using a string you can also define equations and reference existing parameters. For example, the string "d5 / 4" is valid, assuming there is a parameter named d5.

Type As ObjectTypeEnum
> Read-only property returning BIMConduitConnectorDefinitionObject indicating this object's type.

15 BIMDuctConnectorDefinition Object

A programming object that represents the various settings that can be defined for a duct connector. This object doesn't represent the actual connector but only represents the settings and is used in the creation, query, and edit of a connector.

Application As Object
> Read-only property that returns the root Application object.

ConnectionType As BIMDuctConnectionTypeEnum
Read-write property that specifies the connection type for this duct connector.

For a newly created BIMDuctConnectorDefinition object you must first set the SystemType property before setting the value of this property.

Description As String
Read-write property that gets and sets the description of this connector.

For a newly created BIMDuctConnectorDefinition object you must first set the SystemType property before setting the value of this property.

Diameter As Variant
Read-only property that provides access to the connector diameter. When the BIMDuctConnectorDefinition object has been created using the CreateBIMDuctConnectorDefinition method, this property returns a Double indicating the diameter, (in centimeters), of the connector as defined by the input geometry. After the definition object has been used to create a connector, this property returns a parameter that defines the diameter of the connector.

For a newly created BIMDuctConnectorDefinition object you must first set the SystemType property before setting the value of this property.

To change the diameter of an existing connector you can either edit the geometry that's controlling the diameter or set the override diameter using the DiameterOverride property.

DiameterOverride As Variant
Read-write property that gets and sets the diameter override for this connector. If the value returned by this property is an empty string (""), this indicates that there is no override for the diameter. Setting this property to an empty string will remove an override. If there is an override this property will return the expression of the parameter controlling the diameter.

The value can be set using either a double or a string. If a double is input, the units are centimeters. If a string is input, the units can be specified as part of the string or it will default to the current length units of the document. When using a string you can also define equations and reference existing parameters. For example, the string "d5 / 4" is valid, assuming there is a parameter named d5.

For a newly created BIMDuctConnectorDefinition object you must first set the SystemType property before setting the value of this property.

FlowConfiguration As BIMDuctFlowConfigurationEnum
Read-write property that gets and sets how the flow is configured for this connector.

For a newly created BIMDuctConnectorDefinition object you must first set the SystemType property before setting the value of this property.

FlowDirection As BIMFlowDirectionEnum
Read-write property that gets and sets the flow direction for this connector.

For a newly created BIMDuctConnectorDefinition object you must first set the SystemType property before setting the value of this property.

FlowValue As Double
Read-write property that gets and sets the flow rate or flow factor for this connector. If the FlowConfiguration property returns kCalculatedDuctFlowConfigurationType then this property is ignored. If the FlowConfiguration property returns kPresetDuctFlowConfigurationType then this property defines the flow rate. If the FlowConfiguration property returns

---

kSystemDuctFlowConfigurationType then this property defines the flow factor. The units for flow are liters per second.

For a newly created BIMDuctConnectorDefinition object you must first set the SystemType property before setting the value of this property.

Height As Variant

Read-only property that provides access to the connector height. When the BIMDuctConnectorDefinition object has been created using the CreateBIMDuctConnectorDefinition method, this property returns a Double indicating the height, (in centimeters), of the connector as defined by the input geometry. After the definition object has been used to create a connector, this property returns a parameter that defines the height of the connector.

To change the height of an existing connector you can either edit the geometry that's controlling the height or set the override height using the HeightOverride property.

HeightOverride As Variant

Read-write property that gets and sets the height override for this connector. If the value returned by this property is an empty string (""), this indicates that there is no override for the height. Setting this property to an empty string will remove an override. If there is an override this property will return the expression of the parameter controlling the height.

The value can be set using either a double or a string. If a double is input, the units are centimeters. If a string is input, the units can be specified as part of the string or it will default to the current length units of the document. When using a string you can also define equations and reference existing parameters. For example, the string "d5 / 4" is valid, assuming there is a parameter named d5.

LossMethod As BIMDuctLossMethodTypeEnum

Read-write property that gets and sets the loss method defined for this connection.

For a newly created BIMDuctConnectorDefinition object you must first set the SystemType property before setting the value of this property.

LossValue As Double

Read-write property that gets and sets the loss value. If the LossMethod property returns kNoneDuctLossMethodType then this property is ignored. If the LossMethod property returns kCoefficientDuctLossMethodType then this property defines the loss coefficient. If the LossMethod property returns kSpecificLossDuctLossMethodType then this property defines the pressure loss. This units of pressure used are Pascals.

For a newly created BIMDuctConnectorDefinition object you must first set the SystemType property before setting the value of this property.

SystemType As BIMDuctSystemTypeEnum

Read-write property that gets and sets the type of duct system this connection is for.

When setting the values on a newly created BIMDuctConnectorDefinition you must set this property first before you can set the values of any of the other properties.

Type As ObjectTypeEnum

Read-only property returning BIMDuctConnectorDefinitionObject indicating this object's type.

Width As Variant

Read-only property that provides access to the connector width. When the BIMDuctConnectorDefinition object has been created using the

CreateBIMDuctConnectorDefinition method, this property returns a Double indicating the width, (in centimeters), of the connector as defined by the input geometry. After the definition object has been used to create a connector, this property returns a parameter that defines the width of the connector.

To change the width of an existing connector you can either edit the geometry that's controlling the width or set the override width using the WidthOverride property.

WidthOverride As Variant
Read-write property that gets and sets the width override for this connector. If the value returned by this property is an empty string (""), this indicates that there is no override for the width. Setting this property to an empty string will remove an override. If there is an override this property will return the expression of the parameter controlling the width.

The value can be set using either a double or a string. If a double is input, the units are centimeters. If a string is input, the units can be specified as part of the string or it will default to the current length units of the document. When using a string you can also define equations and reference existing parameters. For example, the string "d5 / 4" is valid, assuming there is a parameter named d5.

16 BIMElectricalConnectorDefinition Object

A programming object that represents the various settings that can be defined for an electrical connector. This object doesn't represent the actual connector but only represents the settings and is used in the creation, query, and edit of a connector.

ApparentLoad As Double
Read-write property that gets and sets the apparent load associated with this connector. The units of power are Watts

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

This property is only used when the value of the SystemType property is kPowerBalancedElectricalSystemType and is ignored in all other cases.

ApparentLoadPhase1 As Double
Read-write property that gets and sets the phase 1 apparent load associated with this connector. The units of power are Watts.

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

This property is only used when the value of the SystemType property is kPowerUnbalancedElectricalSystemType and is ignored in all other cases.

ApparentLoadPhase2 As Double
Read-write property that gets and sets the phase 2 apparent load associated with this connector. The units of power are Watts.

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

This property is only used when the value of the SystemType property is kPowerUnbalancedElectricalSystemType and is ignored in all other cases.

ApparentLoadPhase3 As Double

Read-write property that gets and sets the phase 3 apparent load associated with this connector. The units of power are Watts.

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

This property is only used when the value of the SystemType property is kPowerUnbalancedElectricalSystemType and is ignored in all other cases.

Application As Object
Read-only property that returns the root Application object.

Description As String
Read-write property that gets and sets the description of this connector.

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

HasMotor As Boolean
Read-write property that gets and sets whether or not there is a motor associated with this connector.

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

This property is only used when the value of the SystemType property is kPowerBalancedElectricalSystemType or kPowerUnbalancedElectricalSystemType and is ignored in all other cases.

LoadClassification As String
Read-write property that gets and sets the load classification associated with this connector.

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

This property is only used when the value of the SystemType property is kPowerBalancedElectricalSystemType or kPowerUnbalancedElectricalSystemType and is ignored in all other cases.

NumberOfPoles As Long
Read-write property that gets and sets the number of poles associated with this connector. Valid values are 1, 2, or 3.

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

This property is only used when the value of the SystemType property is kPowerBalancedElectricalSystemType or kPowerUnbalancedElectricalSystemType and is ignored in all other cases.

PowerFactor As Double
Read-write property that gets and sets the power factor associated with this connector.

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

This property is only used when the value of the SystemType property is kPowerUnbalancedElectricalSystemType and is ignored in all other cases.

PowerFactorState As BimElectricalPowerFactorStateEnum
Read-write property that gets and sets the power factor associated with this connector.

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

This property is only used when the value of the SystemType property is kPowerUnbalancedElectricalSystemType and is ignored in all other cases.

SystemType As BIMElectricalSystemTypeEnum
Read-write property that specifies the connection type for this duct connector.

When setting the values on a newly created BIMElectricalConnectorDefinition you must set this property first before you can set the values of any of the other properties.

Type As ObjectTypeEnum
Read-only property returning BIMElectricalConnectorDefinitionObject indicating this object's type.

Voltage As double
Read-write property that gets and sets the voltage associated with this connector. The units of voltage is the Volt.

For a newly created BIMElectricalConnectorDefinition object you must first set the SystemType property before setting the value of this property.

This property is only used when the value of the SystemType property is kPowerBalancedElectricalSystemType or kPowerUnbalancedElectricalSystemType and is ignored in all other cases.

17 BIMPipeConnectorDefinition Object

A programming object that represents the various settings that can be defined for a pipe connector. This object doesn't represent the actual connector but only represents the settings and is used in the creation, query, and edit of a connector.

AllowSlopeAdjustment As Boolean
Read-write property that gets and sets whether or not slope adjustment is allowed for this connector.

For a newly created BIMPipeConnectorDefinition object you must first set the SystemType property before setting the value of this property.

Application As Object
Read-only property that returns the root Application object.

ConnectionType As BIMPipeConnectionTypeEnum
Read-write property that gets and sets the connection type for this connector.

For a newly created BIMPipeConnectorDefinition object you must first set the SystemType property before setting the value of this property.

Description As String
Read-write property that gets and sets the description of this connector.

For a newly created BIMPipeConnectorDefinition object you must first set the SystemType property before setting the value of this property.

Diameter As Variant

Read-only property that provides access to the connector diameter. When the BIMPipeConnectorDefinition object has been created using the CreateBIMPipeConnectorDefinition method, this property returns a Double indicating the diameter, (in centimeters), of the connector as defined by the input geometry. After the definition object has been used to create a connector, this property returns a parameter that defines the diameter of the connector.

To change the diameter of an existing connector you can either edit the geometry that's controlling the diameter or set the override diameter using the DiameterOverride property.

DiameterOverride As Variant
Read-write property that gets and sets the diameter override for this connector. If the value returned by this property is an empty string (""), this indicates that there is no override for the diameter. Setting this property to an empty string will remove an override. If there is an override this property will return the expression of the parameter controlling the diameter.

The value can be set using either a double or a string. If a double is input, the units are centimeters. If a string is input, the units can be specified as part of the string or it will default to the current length units of the document. When using a string you can also define equations and reference existing parameters. For example, the string "d5 / 4" is valid, assuming there is a parameter named d5.

FlowConfiguration As BIMPipeFlowConfigurationEnum
Read-write property that gets and sets the loss method used for this connector.

For a newly created BIMPipeConnectorDefinition object you must first set the SystemType property before setting the value of this property.

FlowDirection As BIMPipeFlowDirectionEnum
Read-write property that gets and sets the flow direction for this pipe connector.

For a newly created BIMPipeConnectorDefinition object you must first set the SystemType property before setting the value of this property.

FlowValue As Double
Read-write property that gets and sets the value for the flow. If the value of the FlowConfiguration property is kCalculatedFlowConfigurationType then this value is not used. If the value of the FlowConfiguration property is kPresetFlowConfigurationType then this value defines the flow value where the units for flow are liters per second. If the value of the FlowConfiguration property is kSystemFlowConfigurationType then this value defines the flow factor. If the value of the FlowConfiguration property is kFixtureUnitsFlowConfigurationType then this value defines the fixture units.

For a newly created BIMPipeConnectorDefinition object you must first set the SystemType property before setting the value of this property.

LossMethod As BIMPipeLossMethodEnum
Read-write property that gets and sets the loss method used for this connector.

For a newly created BIMPipeConnectorDefinition object you must first set the SystemType property before setting the value of this property.

LossValue As Double
Read-write property that gets and sets the value for the loss method. If the value of the LossMethod property is kNoneLossMethodType, then this value is not used. If the value of the LossMethod property is kKCoefficientLossMethodType then this value is the K Coefficient

value. If the value of the LossMethod property is kSpecificLossLossMethodType then this value is the pressure drop value where the units for pressure are Pascals.

For a newly created BIMPipeConnectorDefinition object you must first set the SystemType property before setting the value of this property.

NominalDiameter As Variant
Read-write property that provides access to the nominal diameter of the pipe connection. This property is writable when the definition object has been created using the CreateBIMPipeConnectorDefinition method. The value can be set using either a double or a string. If a double is input, the units are centimeters. If a string is input, the units can be specified as part of the string or it will default to the current length units of the document. When using a string you can also define equations and reference existing parameters. For example, the string "d5 * 2" is valid, assuming there is a parameter named d5.

When the BIMPipeConnectorDefinitionobject has been obtained from an existing BimConnector object this property is read-only and returns the Parameter object that defines the nominal diameter. In this case, you can use the properties of the parameter object to change the value.

SystemType As BIMPipeSystemTypeEnum
Read-write property that specifies the connection type for this duct connector.

When setting the values on a newly created BIMPipeConnectorDefinition you must set this property first before you can set the values of any of the other properties.

Type As ObjectTypeEnum
Read-only property returning BIMPipeConnectorDefinitionObject indicating this object's type.

18 BIMConnectorLinks Object

Collection object that provides access to all of the links between connectors and supports the ability to create new links.

Add (*ConnectorOne* As BIMConnector, *ConnectorTwo* As BIMConnector) As BIMConnectorLink
Read-only property that returns the specified BIMConnectorLink object from the collection. The two input connectors must be of the same type and are limited to duct, pipe, and electrical connectors.

Parameter	Type	Tags	Description
ConnectorOne	BIMConnector	In	Input BIMConnector that defines the first of two connectors to be connected by the link.
ConnectorTwo	BIMConnector	In	Input BIMConnector that defines the second of two connectors to be connected by the link.

Application As Object
Read-only property that returns the root Application object.

Count As Long
Read-only property that returns the total number of all types of connector links in this BIMDocument.

Item (*Index* As Variant) As BIMConnectorLink
Read-only property that returns the specified BIMConnectorLink object from the collection.

Parameter	Type	Tags	Description

Index	Variant	In	Input Variant value that specifies the BIM connector link to return. This is the index of the item in the collection where the first item is 1. It can also be the name of the connector. If an out of range value or an unknown name is provided an error will occur.

Type As ObjectTypeEnum
Read-only property returning kBIMConnectorLinksObject indicating this object's type.

19 BIMConnectorLink Object

Object that represents a link between two BIM connectors.

Application As Object
Read-only property that returns the root Application object.

ConnectorOne As BIMConnector
Read-only property that gets the first of two connectors that this link is between.

ConnectorTwo As BIMConnector
Read-only property that gets the first of two connectors that this link is between.

Delete
Method that deletes this link.

GetReferenceKey (*ReferenceKey()* As Byte, Optional *KeyContext* As Long)
Method that generates and returns the reference key for this entity.   A reference key is an array of bytes that can be used as a persistent reference for an entity. To obtain the entity at a later time using the reference key you use the BindKeyToObject method of the object. The ReferenceKeyManager object is obtained using the ReferenceKeyManager property of the Document object.

Parameter	Type	Tags	Description
ReferenceKey	Byte()	Out	Output array of Bytes that contains the reference key.
KeyContext	Long	In	Input Long that specifies the key context. The key context must be supplied when working with any B-Rep entities (and SurfaceBody, FaceShell, Face, Edge, EdgeUse and Vertex objects). A key context is created using the CreateKeyContext method of the ReferenceKeyManager object. For all other object types, the key context argument is not used and is ignored if provided.

Name As String
Read-write property that gets and sets the displayed name of the connector. This is the name that is visible in the browser and is editable by the end-user.

Type As ObjectTypeEnum
Read-only property returning kBIMConnectorLinkObject indicating this object's type.

20 BIMConnectorDefinitionTypeEnum List

Description of the enum.

```
kBIMCableTrayConnectorType
kBIMConduitConnectorType
kBIMDuctConnectorType
kBIMElectricalConnectorType
kBIMPipeConnectorType
```

21 BIMConnectorShapeEnum List

Description of the enum.

```
kCircularShapeConnector
kOvalShapeConnector
kRectangularShapeConnector
kUndefinedShapeConnector
```

22 BIMCableTrayConnectionTypeEnum List

Description of the enum.

```
kElectricallyBondedConnectionType
kUndefinedConnectionType
```

23 BIMConduitConnectionTypeEnum List

Description of the enum.

```
kCompressionConduitConnectionType
kGluedConduitConnectionType
kSetScrewConduitConnectionType
kThreadedConduitConnectionType
kUndefinedConduitConnectionType
```

24 BIMDuctSystemTypeEnum List

Description of the enum.

```
kExhaustDuctSystemType
kOtherDuctSystemType
kReturnDuctSystemType
kSupplyDuctSystemType
kUndefinedDuctSystemType
```

25 BIMDuctLossMethodTypeEnum List

Description of the enum.

```
kNoneDuctLossMethodType
kCoefficientDuctLossMethodType
kSpecificLossDuctLossMethodType
```

26 BIMDuctFlowConfigurationEnum List

Description of the enum.

```
kCalculatedDuctFlowConfigurationType
```

```
kPresetDuctFlowConfigurationType
kSystemDuctFlowConfigurationType
```

27 BIMFlowDirectionEnum List

Description of the enum.

```
kBiDirectionalFlowDirectionType
kInFlowDirectionType
kOutFlowDirectionType
```

28 BIMDuctConnectionTypeEnum List

Description of the enum.

```
kBandedDuctConnectionType
kClippedDuctConnectionType
kFlangeDuctConnectionType
kMasticDuctConnectionType
kOverCollarDuctConnectionType
kRawEdgeDuctConnectionType
kSlipDriveDuctConnectionType
kSlipJointDuctConnectionType
kUndefinedDuctConnectionType
kVanStoneDuctConnectionType
```

29 BIMElectricalSystemTypeEnum List

Various types of electrical systems.

```
kCommunicationElectricalSystemType
kControlsElectricalSystemType
kDataElectricalSystemType
kFireAlarmElectricalSystemType
kNurseCallElectricalSystemType
kPowerBalancedElectricalSystemType
kPowerUnbalancedElectricalSystemType
kSecurityElectricalSystemType
kTelephoneElectricalSystemType
```

30 BimElectricalPowerFactorStateEnum List

Various types of electrical power factor states.

```
kLaggingPowerFactorStateType
kLeadingPowerFactorStateType
```

31 BIMPipeSystemTypeEnum List

Various types of pipe system types.

```
kDomesticColdWaterPipeSystemType
kDomesticHotWaterPipeSystemType
kFireProtectionDryPipeSystemType
```

```
kFireProtectionOtherPipeSystemType
kFireProtectionPreActionPipeSystemType
kFireProtectionWetPipeSystemType
kHydroPonicReturnPipeSystemType
kHydroPonicSupplyPipeSystemType
kOtherPipeSystemType
kSanitaryPipeSystemType
kUndefinedPipeSystemType
```

32 BIMPipeLossMethodEnum List

Various types of pipe loss methods.

```
kNonePipeLossMethodType
kKCoefficientPipeLossMethodType
kSpecificLossPipeLossMethodType
```

33 BIMPipeFlowConfigurationEnum List

Various types of pipe flow configurations.

```
kCalculatedFlowConfigurationType
kPresetFlowConfigurationType
kSystemFlowConfigurationType
kFixtureUnitsFlowConfigurationType
```

34 BIMPipeConnectionTypeEnum List

Various types of pipe connections.

```
kBrazedPipeSystemType
kButtWeldedPipeSystemType
kCapillaryPipeSystemType
kCompressionPipeSystemType
kCouplingPipeSystemType
kCrimpedPipeSystemType
kFlangePipeSystemType
kFusionPipeSystemType
kGluedPipeSystemType
kGroovedPipeSystemType
kSlipJointPipeSystemType
kSocketWeldedPipeSystemType
kSolderedPipeSystemType
kThreadedPipeSystemType
kUndefinedPipeSystemType
```

35 Code Samples

```
' Sample program that will add a connector to a selected circular edge in a part
' document and will export the data as a BIM component.
Public Sub CreateConnectorAndWriteFile()
    Dim partDoc As PartDocument
    Set partDoc = ThisApplication.ActiveDocument

    ' Check that the document has been saved.  Exporting the BIM component
    ' will fail if the document hasn't been saved.
```

```
    If partDoc.fullFilename = "" Then
        MsgBox "The document must be saved before the BIM component can be exported."
        Exit Sub
    End If

    ' Have the user pick a circular edge.
    Dim circEdge As Edge
    Set circEdge = ThisApplication.CommandManager.Pick( _
                    SelectionFilterEnum.kPartEdgeCircularFilter, _
                    "Select a circular edge")

    ' Get the BIM component from the part document.
    Dim BIMComp As BIMComponent
    Set BIMComp = partDoc.ComponentDefinition.BIMComponent

    ' Package the geometry into an object collection.
    Dim geom As ObjectCollection
    Set geom = ThisApplication.TransientObjects.CreateObjectCollection
    Call geom.Add(circEdge)

    ' Create the definition.
    Dim pipeDef As BIMPipeConnectorDefinition
    Set pipeDef = BIMComp.Connectors.CreatePipeConnectorDefinition(geom, _
                                BIMConnectorShapeEnum.kCircularShapeConnector)

    ' Override the diameter. You can use a value or a string,
    ' as shown here, that is an expression.
    pipeDef.DiameterOverride = ".25 in"

    ' Set the various pipe properties, making sure to set the system type first.
    pipeDef.SystemType = BIMPipeSystemTypeEnum.kHydroPonicSupplyPipeSystemType
    pipeDef.ConnectionType = BIMPipeConnectionTypeEnum.kThreadedPipeConnectionType
    pipeDef.FlowConfiguration = _
                        BIMPipeFlowConfigurationEnum.kPresetFlowConfigurationType
    pipeDef.FlowDirection = BIMFlowDirectionEnum.kInFlowDirectionType
    pipeDef.FlowValue = 0.631
    pipeDef.LossMethod = BIMPipeLossMethodEnum.kSpecificLossPipeLossMethodType
    pipeDef.LossValue = 5
    pipeDef.Description = "Single pipe inlet"

    ' Create the cable tray definition.
    ' THE NAME CANNOT HAVE ANY SPACES AND MUST BE UNIQUE!!
    Dim pipeConnector As BIMConnector
    Set pipeConnector = BIMComp.Connectors.Add(pipeDef, "MyPipeConnector")

    ' Set the component description, as defined in the file:
    ' [Design Data]\BIM Exchange\Support\AuthoringTemplates\Categories.xml
    BIMComp.ComponentDescription.ComponentType = "23.75.70.17.21"

    ' Export the component.
    Call BIMComp.ExportBuildingComponent("C:\Temp\TestExport.adsk")
End Sub
```