



## **Application Objects API Reference Manual**

**June 1<sup>st</sup>, 2000**

Vince Matacale  
Software Engineer

© Copyright 2000 United States Data Corporation. All rights reserved.

NOTICE:

The information contained in this document (and other media provided herewith) constitutes confidential information of United States Data Corporation (“USDATA”) and is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. Such information is not to be disclosed, used or copied by, or transferred to, any individual, corporation, company or other entity, in any form, by any means or for any purpose, without the express written permission of USDATA.

The information contained in this document and related media constitutes documentation relating to a software product and is being provided solely for use with such software product. The software product was provided pursuant to a separate license or other agreement and such information is subject to the restrictions and other terms and conditions of such license or other agreement.

The information contained in this document and related media is subject to change without notice and does not represent a commitment on the part of USDATA. Except for warranties, if any, set forth in the separate license or other agreement relating to the applicable software product, USDATA makes no warranty, express or implied, with respect to such information or such software product.

USDATA and FactoryLink are registered trademarks of United States Data Corporation in the United States and/or other countries. Open Software Bus is a registered trademark licensed to United States Data Corporation. All other brand or product names are trademarks or registered trademarks of their respective holders.

<b>INTRODUCTION</b>	<b>5</b>
Purpose:	5
Overview:	5
Recommended Usage:	5
<b>APPLICATION OBJECTS OBJECT LIBRARY</b>	<b>6</b>
<b>FIGURE 1 - APPLICATION OBJECTS HIERARCHY</b>	<b>6</b>
<b>AOBJECTSERVER</b>	<b>7</b>
Properties:	7
Methods:	8
Events:	37
<b>IOBJECTS</b>	<b>39</b>
Properties:	39
Methods:	39
<b>IOBJECT</b>	<b>41</b>
Properties:	41
Methods:	42
<b>INSTANCES</b>	<b>43</b>
Properties:	43
Methods:	43
<b>INSTANCE</b>	<b>45</b>
Properties:	45
Methods:	46
<b>AOTVARINSTANCES</b>	<b>47</b>
Properties:	47
Methods:	47
<b>AOTVARINSTANCE</b>	<b>49</b>
Properties:	49
<b>AOTVARIABLES</b>	<b>50</b>
Properties:	50
Methods:	50
<b>AOTVARIABLE</b>	<b>52</b>
Properties:	52

<b>Methods:</b>	<b>53</b>
<b>AOPROPERTIES</b>	<b>55</b>
<b>Properties:</b>	<b>55</b>
<b>Methods:</b>	<b>55</b>
<b>AOPROPERTY</b>	<b>57</b>
<b>Properties:</b>	<b>57</b>
<b>AOOPCITEMS</b>	<b>58</b>
<b>Properties:</b>	<b>58</b>
<b>Methods:</b>	<b>58</b>
<b>AOOPCITEM</b>	<b>60</b>
<b>Properties:</b>	<b>60</b>
<b>AOOBJECTS</b>	<b>61</b>
<b>Properties:</b>	<b>61</b>
<b>Methods:</b>	<b>61</b>
<b>AOOBJECT</b>	<b>62</b>
<b>Properties:</b>	<b>62</b>
<b>Methods:</b>	<b>68</b>
<b>AOIOBJDATA</b>	<b>71</b>
<b>Properties:</b>	<b>71</b>

## Introduction

### **Purpose:**

Application Objects consists of a hierarchy of objects grouped together in a component object model (COM) dynamically linked library (dll) named ApplicationObjects.dll. The purpose of this document is to outline the functionality of each of these objects as well as the way in which they fit together. When applicable, this document refers to other objects and sub-systems that are provided with FactoryLink 7. These objects and sub-systems are mentioned only to the extent of their relevance to Application Objects.

### **Overview:**

Typically, Application Objects are programmatically used by getting an AObjectServer object, and then invoking methods on it to perform actions, such as creating instances of an Application Object or looking up and returning a collection of Application Objects Classes.

### **Recommended Usage:**

The objects contained in ApplicationObjects.dll may change. It is recommended that programmers do not early bind to these objects. Early binding in Visual Basic is the process of adding a reference to a dll and then using the objects it provides. All Application Objects support an interface in Flink7Defns.tlb that can be guaranteed not to change. It is recommended that programmers early bind to the type library Flink7Defns.tlb and then create their objects using the ProgID. The ProgID of each object is ApplicationObjects.*ObjectName* where *ObjectName* is the name of the object to be created. In Figure 1, the Application Objects Hierarchy, the name of the each object is given in the top and the name of the Flink7Defns interface it supports is given below it in parentheses. This example creates a AObjectServer Object using its Flink7Defns interface:

```
Dim mAOServer as IAOServer
```

```
Set mAOServer = CreateObject("ApplicationObjects.AObjectServer")
```

This example requires a reference to Flink7Defns.tlb (FactoryLink 7 Object Definitions).

Sometimes it will be necessary to early bind to an object. Usually, this is when the object publishes events. In Application Objects, the AObjectServer (IAOServer) is the only object that publishes events. If these are required, it will be necessary to early bind to the object. The above example would be rewritten as:

```
Dim withevents mAOServer as AObjectServer
```

```
Set mAOServer = new AObjectServer
```

This example requires a reference to ApplicationObjects.dll (FactoryLink 7 Object Definitions).

Note that subsequent releases of Application Objects may require that you modify and recompile your program when you early bind to ApplicationObjects.dll.

# Application Objects Object Library

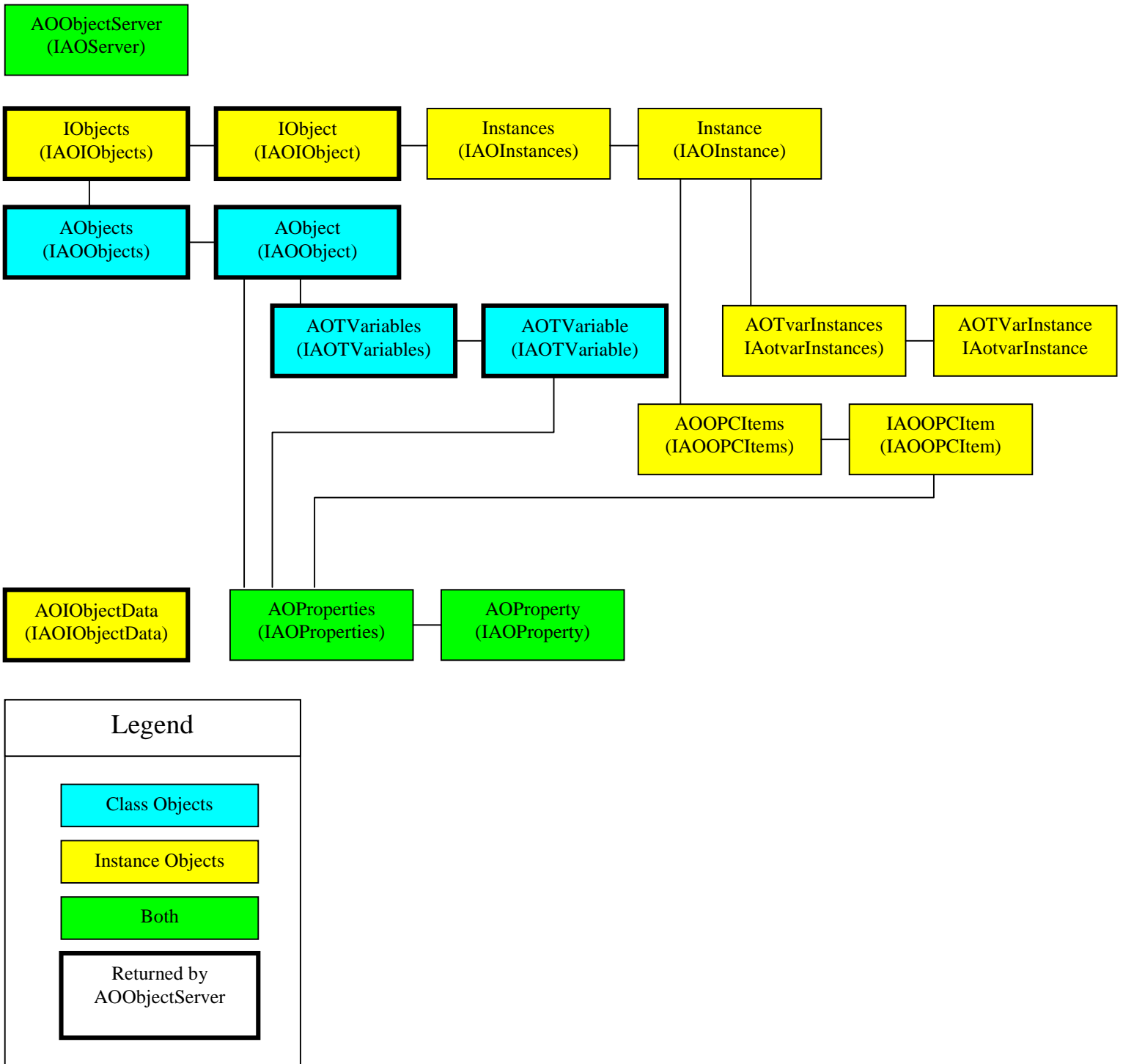


Figure 1 - Application Objects Hierarchy

## AObjectServer

AObjectServer is the main object that contains the basic functionality of Application Objects. It is generally used to return other Application Objects and/or to perform operations on them.

Flink7Defns Interface: IAOServer

Public: Yes

Creatable: Yes

### **Properties:**

SilentMode	<p>A Boolean property which dictates whether the server displays error messages or raises errors. This property has not yet been implemented.</p> <p>Type: Boolean</p> <p>Access: Read/Write</p>
ParentHwnd	<p>A long that is the parent hwnd of the window which is controlling Application Objects. This property is used to control the dialogs that Application displays.</p> <p>Type: Long</p> <p>Access: Read/Write</p>
TagEdit	<p>A property of type EnumFL6TagExistOptions. The value of this property determines what happens when a config object wants to create a tag, and the tag to be created already exists, but with different properties. For example, if tagA is going to be created as an Analog and it already exists as a Float.</p> <p>By default, when an object is created, the value is FL6TagAbortIfDiff. This will cause a prompt to appear allowing the user to choose to keep the tag (FL6TagKeepExisting ) or to modify the tag (FL6TagModifyExisting). This choice can be applied to all objects if desired. The property is reset to FL6TagAbortIfDiff when the ApplicationObject has finished unless the user has explicitly set it.</p> <p>Type: Enumeration EnumFL6TagExistOptions</p> <p>FL6TagAbortIfDiff = 0 FL6TagKeepExisting = 1 FL6TagModifyExisting = 2</p> <p>Access: Read/Write</p>
AttributeModFlags	<p>This property is used in conjunction with the TagEdit property to set more complicated rules for tag modification. By default, a dialog is presented to the user that will set this property.</p> <p>The property is reset to 0 when the ApplicationObject has finished unless the user has explicitly set it.</p>

Note that this property has no effect if the TagEdit property is not set to FL6TagModifyExisting (2)

Rules:

- 1 Modify the Domain
- 2 Modify the Type
- 4 Modify the Dimension
- 8 Modify the dimension only if it would make it larger
- 16 Modify the Description
- 32 Modify the Description only if it is blank
- 64 Modify the persistence settings
- 128 Modify the persistence change status settings
- 256 Modify the default value
- 512 Modify the default value only if it is blank
- 1024 Modify the Message Length
- 2048 Modify the Message Length only if it would make it larger

Type: Long

Access: Read/Write

**Methods:**

Create                      Creates an Application Object, AObject, initializes it's properties, and returns it to the user. Note that this does not actually save the object to the database. See the Update method for information on how to save an Application Object.

Parameters:

- |            |   |
|------------|---|
| Flapp      | The FactoryLink Application Directory.<br><br>Type: String<br><br>Access: In  |
| Name       | Optional string Parameter that is used to set the name property of the AObject object returned.<br><br>Type: String<br><br>Access: In   |
| ObjectType | Optional AOAppObjectType parameter that is used to set the ObjectType property of the AObject object returned. This property determines what the Properties collection on the AObject is filled with. The default is AOApplicationObject. |



Type: Enumeration AOAppObjectType

AOFlinkRecord = 0  
AOFlinkFile = 1  
AOApplicationObject = 2

Access: In

DefaultNumberInstances Optional Parameter that is used to set the DefaultNumberInstances property of the AObject object returned.

Type: Long

Access: In

InstanceBaseName Optional Parameter that is used to set the InstanceBaseName property of the AObject object returned.

Type: String

Access: In

InstanceDescription Optional Parameter that is used to set the InstanceDescription property of the AObject object returned.

Type: String

Access: In

InstanceKey Optional Parameter that is used to set the InstanceKey property of the AObject object returned.

Type: String

Access: In

InstanceDisableFlagField Optional Parameter that is used to set the InstanceDisableFlagField property of the AObject object returned.

Type: String

Access: In

GroupDescription Optional Parameter that is used to set the GroupDescription property of the AObject object returned.

Type: String

Access: In

DisableVariable

Optional Parameter that is used to set the DisableVariable property of the AObject object returned.

Type: String

Access: In

IAddressOffset

Optional Parameter that is used to set the IAddressOffset property of the AObject object returned.

Type: Long

Access: In

IAddressMultiplier

Optional Parameter that is used to set the IAddressMultiplier property of the AObject object returned.

Type: Long

Access: In

Return Value: AObject

GetAObject

Returns an Application Object, AObject based on either name or ID.

Parameters:

Flapp

The FactoryLink Application Directory.

Type: String

Access: In

ID

Optional Parameter that is used to find the AObject being returned. Either the ID or Name parameters must be passed into the call.

Type: String

Access: In

Name Optional Parameter that is used to find the AObject being returned. Either the ID or Name parameters must be passed into the call.

Type: String

Access: In

Return Value: AObject

UpdateAObject

Saves the Application Object into the database tables. Either creates a new Application Object or updates an existing object. It returns the Application Object that was just updated.

Parameters:

AO The Application Object being updated or created.

Type: String

Access: In/Out

Return Value: AObject

DeleteObject

Deletes an Application Object from the database, based on either name or ID. The object must not be used by other objects or the call will fail

Parameters:

Flapp The FactoryLink Application Directory.

Type: String

Access: In

ID Optional Parameter that is used to find the AObject being deleted. Either the ID or Name parameters must be passed into the call.

Type: String

Access: In

Name Optional Parameter that is used to find the AObject being deleted. Either the ID or

Name parameters must be passed into the call.

Type: String

Access: In

Return Value: None

ReturnTVVariables

Retrieves an AOTVariables containing all template variables used by an Application Object.

Parameters:

FlappDir

The FactoryLink Application Directory.

Type: String

Access: In

ParentID

Optional, the ID of the Application Object. If this parameter is omitted, then all template variables in the Application are returned.

Type: String

Access: In

Return Value: AOTVariables

AddTVVariableToDatabase Creates or updates a Template Variable in the Application Object database. The New Variable AOTVariable is returned.

Parameters:

Flapp

The FactoryLink Application Directory.

Type: String

Access: In

ID

The ID of the Template Variable. This parameter should be an empty string for new template variables

Type: String

	Access: In
Name	The Name of the Template Variable. Type: String
	Access: In
Value	The Value of the Template Variable. This parameter is obsolete and should be an empty string. Type: String
	Access: In
DataSourceString	A Packed String representing how the template variable gets its values. The Packed string is obtained by creating a <code>FlinkCDBObjects.cTemplateReplication</code> object (Interface <code>ItemplateReplication</code> ). Once the Replication Properties are set on the object, the <code>ReplicationString</code> property will return the packed string.
PresentationString	A Packed String representing how the template variable is presented to the user. The Packed string is obtained by creating a <code>FlinkCDBObjects.cTemplatePresentation</code> object (Interface <code>ItemplatePresentation</code> ). Once the Presentation Properties are set on the object, the <code>PresentationString</code> property will return the packed string. Type: String
	Access: In
Description	The Description of the Template Variable. Type: String
	Access: In

Return Value: AOTVariable

FindTVVariable Finds and returns a Template Variable from an ID or a Name.

Parameters:

FlappDir	The FactoryLink Application Directory. Type: String
----------	--

		Access: In
	TVariableID	Optional, the ID of the Template Variable. Either the Name or the ID must be present.  Type: String
		Access: In
	TVariableName	Optional, the Name of the Template Variable. Either the Name or the ID must be present.  Type: String
		Access: In
	Return Value: AOTVariable	
ReturnChildren	Returns the child Application Objects an Application Object contains. System Defined Ids may also be used to return all the Application Objects in the system.	
	Parameters:	
	Flapp	The FactoryLink Application Directory.  Type: String  Access: In
	Container	The ID of the Container Application Object. Currently, only objects of type AOApplicationObject(2) can act as containers.  In addition to accepting a valid Application Object ID, the system defined ID, "ApplicationObjectsRoot", can be used which is the root level container that contains all objects.  Type: String  Access: In
	ID	The ID of the Application Object whose children will be returned. This object must exist within the container. In addition to accepting a valid Object ID, the following constants can be used to specify all the objects in the system.

“ApplicationObjectsConfigObjRoot” returns all config objects in the system.

“ApplicationObjectsAppObjRoot” returns all Application objects in the system.

“ApplicationObjectsFileObjRoot” returns all Application objects in the system.

Type: String

Access: In

#### Member

Optional, a qualifier used to differentiate between multiple objects of the same type. By default, the member parameter is an empty string. If there are 2 instances of the same Object inside of an Application Object, the member value of “” would return the children of the first instance and a “2” would return the children of the second object.

Type: String

Access: In

Return Value: AObjects

### AddChildObject

Adds an Object to the Container Object.

Parameters:

#### Flapp

The FactoryLink Application Directory.

Type: String

Access: In

#### ContainerID

The ID of the Container Application Object. This object must exist. Currently, only objects of type AOApplicationObject(2) can act as containers.

Type: String

Access: In

#### ParentID

The ID of the Application Object that will be the parent of the new object. This object must exist. When adding objects to the root level of the Application Object, this ID will be the same as the Container ID.

		Type: String
		Access: In
	ParentMember	Optional, a qualifier used to differentiate between multiple objects of the same type. By default, the member parameter is an empty string. If there are 2 instances of the same Object inside of an Application Object, the member value of "" would refer to the first object and "2" would refer to the second object.
		Type: String
		Access: In
	ChildID	The ID of the Application Object that is being added to the container. This object must exist.
		Type: String
		Access: In
	ChildMember	Optional, a qualifier used to differentiate between multiple objects of the same type. By default, the member parameter is an empty string. If there are 2 instances of the same Object inside of an Application Object, the member value of "" would refer to the first object and "2" would refer to the second object.
		Type: String
		Access: In
	Return Value: None	
SetChildObjects	Adds Sub Objects to the Container Object. This method is obsolete. The AddChildObject method should be used.	
	Parameters:	
	AO	The Application Object the objects will be added to.
		Type: (AObject)
		Access: In/Out
	ParentChildIDList	An variant Array dimensioned as Arr(4,n) where n is the number of child objects to



add. The first element is the ParentID, the second element is the ParentMember, the third element is the Child ID parameter, and the fourth element is the Child member.

Note that the second element is dimensioned from 0 to n where the 0<sup>th</sup> element is ignored.

See AddChildObject.

Type: Variant

Access: In

Return Value: None

ReturnDescendentIDs Returns a variant array of the Ids for all of the descendents of an object.

Parameters:

AO

The Application Object to retrieve the descendents for. This should be a container Application Object. Currently, only objects of type AOApplicationObject(2) can act as containers.

Type: (AOObject)

Access: In/Out

ContainersOnly Optional, a Boolean specifying whether only container objects will be returned. The default is false.

The Return value will be a variant array dimensioned as Array(n) where n will be the number of descendents. The 0<sup>th</sup> element is ignored in the array.

Type: Boolean

Access: In

Return Value: Variant

TVarUsed

Returns a boolean specifying whether the variable is used by any Application.

Parameters:

Flapp

The FactoryLink Application Directory.

Type: String

Access: In

VariableID

Optional, the ID of the Template Variable. Either the Name or the ID must be present.

		Type: String
		Access: In
	VariableName	Optional, the Name of the Template Variable. Either the Name or the ID must be present.
		Type: String
		Access: In
	Return Value: Boolean	
DeleteTVar	Deleted a template variable a boolean specifying whether the variable is used by any Application. The variable must not be used by any objects. Returns true if the delete was successful.	
	Parameters:	
	Flapp	The FactoryLink Application Directory.
		Type: String
		Access: In
	VariableID	Optional, the ID of the Template Variable. Either the Name or the ID must be present.
		Type: String
		Access: In
	VariableName	Optional, the Name of the Template Variable. Either the Name or the ID must be present.
		Type: String
		Access: In
	Return Value: Boolean	
AObjectUsed	Returns a boolean specifying true if the object is used in any object.	
	Parameters:	
	Flapp	The FactoryLink Application Directory.
		Type: String
		Access: In

	ObjectID	Optional, the ID of the Object. Either the Name or the ID must be present.  Type: String  Access: In
	ObjectName	Optional, the Name of the Object. Either the Name or the ID must be present.  Type: String  Access: In
	Return Value:	Boolean
RegisterForEvents		Returns an event object of type AOEvent that will receive Application Object Events. In VB, AOEvent must be declared with the keyword withevents in order to receive Application Object Events.  Parameters:  Return Value: IDispatch
ReturnIChildren		Returns an IObject that contains all of the Instantiated Child Objects.  Parameters:
	Flapp	The FactoryLink Application Directory.  Type: String  Access: In
	ID	The Instance ID of the Instantiated Object.  Type: String  Access: In
	InstanceNumber	The Instance Number of the Name of the Object. Either the Name or the ID must be present.  Type: String  Access: In
	Return Value:	IObjects
ModifyIObject		Takes an IObject that has been modified. Each IAOTVarInstance in its IAOTVarInstances property has an IAOTVarInstances object. This object has a collection of IAOTVarInstance objects. The newvalue property of each was initially set to

Chr\$(0) when the object was retrieved. To modify the object, set the newvalue to the desired value. The ModifyIObject call looks for newvalues that are not chr\$(0). When it finds one, it modifies the Application Object Instance.

The Instance ID of the top most Application Object Instance is returned.

Parameters:

AOIObj	The IObject to be modified. Type: IObject Access: In/Out
ProvideFeedback	An optional parameter which, if set to true, will cause the feedback events to be fired. The default value is true. Type: Boolean Access: In

Return Value: String

ReturnIObjectData

Returns an IAOIObjectData object based on the Flapp, the InstanceID, the Instance Number, and the object type. If these parameters are not specified, it gets the information from the IAOIObject.

Parameters:

InstanceNumber	The Instance Number to be used to look up the information. Type: Long Access: In
AOIObj	Optional, the IObject to be used to look up the information. This information can be passed in directly instead. Type: IObject Access: In
Flapp	Optional, The FactoryLink Application Directory. If this is not used, it will be retrieved from the AOIObj object. Type: String Access: In
InstanceID	Optional The Instance ID of the Instantiated Object. If this is not used, it will be retrieved from the AOIObj object.

Type: String

Access: In

ObjectType

Optional the object type of the Instantiated Object. If this is not used, it will be retrieved from the AOIObj object.

Type: String

Access: In

Return Value: IAIOObjectData

CopyTemplateVariable Copies a template variable. This can be between flapps or within the same flapp. The new variable is returned.

Parameters:

SourceFlapp

The FactoryLink Application Directory that contains the source Template Variable.

Type: String

Access: In

TVariableID

Optional The Template Variable ID of the source template variable. If this is not specified, the TvariableName must be.

Type: String

Access: In

TVariableName

Optional The Template Variable name of the source template variable. If this is not specified, the TvariableID must be.

Type: String

Access: In

DestinationName

Optional The new Template Variable name. If this is not specified, a unique name will be will be generated.

Type: String

Access: In

DestinationFlapp

Optional The destination FactoryLink Application Directory. If this is not specified, it is assumed to be the same as the source.

Type: String

Access: In

Return Value: IAOTVariable

**CopyApplicationObject** Copies an Application Object. This can be between flapps or within the same flapp. Any objects or template variables contained by the source object or any of its children are also copied. The New Object is returned.

Parameters:

**SourceFlapp** The FactoryLink Application Directory that contains the source Object.

Type: String

Access: In

**AOID** Optional The Object ID of the source object. If this is not specified, the AOName must be.

Type: String

Access: In

**AOName** Optional The Object name of the source object. If this is not specified, the AOID must be.

Type: String

Access: In

**DestinationName** Optional The new Object name. If this is not specified, a unique name will be will be generated.

Type: String

Access: In

**DestinationFlapp** Optional The destination FactoryLink Application Directory. If this is not specified, it is assumed to be the same as the source.

Type: String

Access: In

Return Value: IAObject

**RenameTemplateVariable** Renames a template variable. If Successful, true is returned, otherwise false is returned. All configuration containing this template variable will be updated.

Parameters:

Flapp	The FactoryLink Application Directory that contains the Template Variable to be renamed.  Type: String  Access: In
TVariableID	The Template Variable ID of the source template variable.  Type: String  Access: In
NewName	The new template variable name. This name should not be used by a different template variable.  Type: String  Access: In

Return Value: Boolean

**ReturnApplicationConstants** Obsolete. Constants are no longer supported

**SetApplicationConstantValue** Obsolete. Constants are no longer supported

**GetApplicationConstantValue** Obsolete. Constants are no longer supported

**GetIOObject** Looks up and returns an Application Object Instance Object (IOObject).

Parameters:

Flapp	The FactoryLink Application Directory that contains the Instance to be retrieved.  Type: String  Access: In
-------	---

InstanceID	<p>Optional, the Instance ID of the object. If this is not specified, the ObjectName must be.</p> <p>Type: String</p> <p>Access: In</p>
ObjectName	<p>Optional, the Instance name of the object. If this is not specified, the InstanceID must be. The name was stored based on the GroupBaseName property of the Application Object.</p> <p>Type: String</p> <p>Access: In</p>
ParentID	<p>Optional, If this parameter is specified, the ParentInstanceID property of the IAObject returned will be set to this value, otherwise ParentInstanceID will be set to “?”.</p> <p>Type: String</p> <p>Access: In</p>
ParentNumber	<p>Optional, If this parameter is specified, the ParentInstanceNumber property of the IAObject returned will be set to this value, otherwise ParentInstanceID will be set to 0.</p> <p>Type: String</p> <p>Access: In</p>
IncludeInstanceData	<p>Optional, If the value is true, the IAInstances object is filled with instance data otherwise it is left uninitialized.</p> <p>The property InstanceDataAvailable on the IAObject will be set to true if IncludeInstanceData is true.</p> <p>The default is true. This value should be set to false if instance data is not needed.</p> <p>Type: Boolean</p> <p>Access: In</p>
IncludeTVarData	<p>Optional, If the value is true, the IAOTVarInstances object in each IAInstance object in the IAInstances property of the IObject returned is filled</p>



with template variable data otherwise it is left uninitialized.

The property InstanceTVarDataAvailable on the IAObject will be set to true if IncludeTVarData is true.

The default is true. This value should be set to false if Template Variable data is not needed. The IncludeInstanceData parameter must be true to set IncludeTVarData to true.

Type: Boolean

Access: In

IncludeOPCItemData

Optional, If the value is true, the IAOPCItems object in each IAOInstance object in the IAInstances property of the IObject returned is filled with OPC Item data otherwise it is left uninitialized.

The property OPCItemDataAvailable on the IAObject will be set to true if IncludeOPCItemData is true.

The default is true. This value should be set to false if OPC Item data is not needed. The IncludeInstanceData must be true if the IncludeOPCItemData is true.

Type: Boolean

Access: In

Return Value: IObject

CreateIObject

Creates Application Objects Instances in the target Flapp. If successful, the instance ID of the top most object created is returned.

Parameters:

Flapp

The FactoryLink Application Directory to create Application Object instances in.

Type: String

Access: In

ClassName

Optional, the classname of the object. If this is not specified, the ClassID must be.

Type: String

		Access: In
	ClassID	Optional, the Class ID of the object. If this is not specified, the ClassName must be.  Type: String
		Access: In
	NumberOfInstances	Optional, the Number of instances to create. If this not specified, the NumberOfInstances property of the IAObject is used.  Type: String
		Access: In
	ParentPath	Optional, The table that acts as the top most parent record. This is used for the parent record of Config Objects. The notation of a path is FlappDir/Task/Domain/Table1!FieldInfo/ Table2!FieldInfo  Note that forward slashes must be used for the path and back slashes must be used in the Flapp Directory.  Type: String
		Access: In
	ProvideFeedback	An optional parameter which, if set to true, will cause the feedback events to be fired. The default value is true.  Type: Boolean
		Access: In
		Return Value: String
AddObjectInstances		Creates Application Objects Instances in the target Flapp. If successful, the instance ID of the top most object added to is returned.
		Parameters:
	Flapp	The FactoryLink Application Directory that contains the Instance Object to be added to.  Type: String

	Access: In
InstantiatedObj	Optional, the IObject to add to. If this is not specified, either the Objectname or the InstanceID must be.  Type: IObject
	Access: In
InstanceID	Optional, the Instance ID of the object to be added to. If this is not specified, either the InstantiatedObj or the Objectname must be.  Type: String
	Access: In
ObjectName	Optional, the Object Name of the object, this comes from the Group Base Name Property of the Class. If this is not specified, either the InstantiatedObj or the InstanceID must be.  Type: String
	Access: In
ProvideFeedback	An optional parameter which, if set to true, will cause the feedback events to be fired. The default value is true.  Type: Boolean
	Access: In

Return Value: String

InstanceIDFromQualifiedObjectName Returns the InstanceID from a fully Qualified name.

Parameters:

Flapp	The FactoryLink Application Directory that contains the Instance Object.  Type: String
	Access: In

	<p>ObjectName</p>	<p>The Qualified Name of the Object, IObject to add to. The notation GroupBase1.InstanceKey1. GroupBase2.InstanceKey2. Alternatively, the notation GroupBase1(InstanceNumber1). GroupBase2(InstanceNumber2) can be used.</p> <p>Type: String</p> <p>Access: In</p>
	<p>Return Value: String</p>	
<p>CreateIObjectInstance</p>	<p>Creates a single Application Object instance. The Application Object Instance Object must already exist and have at least one instance. If the call succeeds, the Instance ID of the Object will be returned.</p>	
	<p>Parameters:</p>	
	<p>Flapp</p>	<p>The FactoryLink Application Directory that contains the Instance Object to be added to.</p> <p>Type: String</p> <p>Access: In</p>
	<p>ObjectName</p>	<p>Optional, the Object Name of the object, this comes from the Group Base Name Property of the Class. If this is not specified, the InstanceID must be.</p> <p>Type: String</p> <p>Access: In</p>
	<p>InstanceID</p>	<p>Optional, the Instance ID of the object to be added to. If this is not specified, the Objectname must be.</p> <p>Type: String</p> <p>Access: In</p>
	<p>InstanceNumber</p>	<p>The Instance Number to create. If this already exists or is 0, the call will fail.</p> <p>Type: Long</p> <p>Access: In</p>

ProvideFeedback An optional parameter which, if set to true, will cause the feedback events to be fired. The default value is true.

Type: Boolean

Access: In

Return Value: String

DeleteObject

Deletes an entire Application Object instance and all it's children. The Application Object Instance Object must already exist and have at least one instance. If the call succeeds, the Instance ID of the Object will be returned.

Parameters:

Flapp The FactoryLink Application Directory that contains the Instance Object to be added to.

Type: String

Access: In

InstantiatedObj Optional, the IObject to be Deleted. If this is not specified, either the Objectname or the InstanceID must be.

Type: IObject

Access: In

InstanceID Optional, the Instance ID of the object to be Deleted. If this is not specified, either the InstantiatedObj or the Objectname must be.

Type: String

Access: In

ObjectName Optional, the Object Name of the object to be deleted, this comes from the Group Base Name Property of the Class. If this is not specified, either the InstantiatedObj or the InstanceID must be.

Type: String

Access: In

ProvideFeedback An optional parameter which, if set to true, will cause the feedback events to be fired. The default value is true.

Type: Boolean

Access: In

Return Value: None

DeleteIOBJECTINSTANCE Deletes a single Application Object instance. The Application Object Instance Object must already exist.

Parameters:

Flapp The FactoryLink Application Directory that contains the Instance Object to be deleted.

Type: String

Access: In

ObjectName Optional, the Object Name of the object, this comes from the Group Base Name Property of the Class. If this is not specified, the InstanceID must be.

Type: String

Access: In

InstanceID Optional, the Instance ID of the object to be added to. If this is not specified, the Objectname must be.

Type: String

Access: In

InstanceNumber The Instance Number to Delete.

Type: Long

Access: In

ProvideFeedback An optional parameter which, if set to true, will cause the feedback events to be fired. The default value is true.

Type: Boolean

Access: In

Return Value: String

**RemoveObjectInstances** Removes Application Object instances from an Application Object Instance Object. Instance are removed from the end of the object. If the number of instances to be removed is more than the number of instances, the object is completely removed.

Parameters:

Flapp	The FactoryLink Application Directory that contains the Instance Object to be added to.  Type: String  Access: In
NumberOfInstances	The Number of instances to be removed.  Type: String  Access: In
InstantiatedObj	Optional, the IObject to be Deleted from. If this is not specified, either the Objectname or the InstanceID must be.  Type: IObject  Access: In
InstanceID	Optional, the Instance ID of the object to be Deleted from. If this is not specified, either the InstantiatedObj or the Objectname must be.  Type: String  Access: In
ObjectName	Optional, the Object Name of the object to be deleted from. This comes from the Group Base Name Property of the Class. If this is not specified, either the InstantiatedObj or the InstanceID must be.  Type: String  Access: In
ProvideFeedback	An optional parameter which, if set to true, will cause the feedback events to be fired. The default value is true.  Type: Boolean  Access: In

Return Value: None

## RefreshIObjct

Refreshes all configuration by reading the data sources again and updating all objects. Note that Prompt Template variables are not prompted. These must be changed through the ModifyIObjct method.

### Parameters:

Flapp	The FactoryLink Application Directory that contains the Instance Object to be refreshed.  Type: String  Access: In
InstantiatedObj	Optional, the IObjct to be refreshed. If this is not specified, either the Objectname or the InstanceID must be.  Type: IObjct  Access: In
InstanceID	Optional, the Instance ID of the object to be refreshed. If this is not specified, either the InstantiatedObj or the Objectname must be.  Type: String  Access: In
ObjectName	Optional, the Object Name of the object to be refreshed. This comes from the Group Base Name Property of the Class. If this is not specified, either the InstantiatedObj or the InstanceID must be.  Type: String  Access: In
ProvideFeedback	An optional parameter which, if set to true, will cause the feedback events to be fired. The default value is true.  Type: Boolean  Access: In

Return Value: None



**RefreshObjectInstance** Refreshes all configuration in a single instance by reading the data sources again and updating all objects. Note that Prompt Template variables are not prompted. These must be changed through the **ModifyObject** method.

Parameters:

<b>Flapp</b>	The FactoryLink Application Directory that contains the Instance Object to be refreshed.  Type: String  Access: In
<b>InstantiatedObj</b>	Optional, the IObject to be refreshed. If this is not specified, either the Objectname or the InstanceID must be.  Type: IObject  Access: In
<b>InstanceID</b>	Optional, the Instance ID of the object to be refreshed. If this is not specified, either the InstantiatedObj or the Objectname must be.  Type: String  Access: In
<b>ObjectName</b>	Optional, the Object Name of the object to be refreshed. This comes from the Group Base Name Property of the Class. If this is not specified, either the InstantiatedObj or the InstanceID must be.  Type: String  Access: In
<b>InstanceNumber</b>	The instances number to be refreshed.  Type: Long  Access: In
<b>ProvideFeedback</b>	An optional parameter which, if set to true, will cause the feedback events to be fired. The default value is true.  Type: Boolean  Access: In

Return Value: None

QualifiedObjectNameFromInstanceID Returns a fully Qualified name from an InstanceID.

Parameters:

Flapp	The FactoryLink Application Directory that contains the Instance Object.  Type: String  Access: In
InstanceID	The Instance ID of the object.  Type: String  Access: In
NameSyntax	The syntax to format the name in. Values can be AOINInstanceName(0), which returns GroupBase1.InstanceKey1. GroupBase2.InstanceKey2 or AOINInstanceArray(2) which returns GroupBase1(InstanceNumber1). GroupBase2(InstanceNumber2) can be used.  Type: Enumeration AOINNameSyntax  Access: In
InstanceNumber	The instances number of the last object. If this is not specified, the name ends with an GroupBase otherwise, it ends with an instance number or key.  Type: Long  Access: In

Return Value: String

ReturnInstantiatedRootObjects Returns all root Instance Objects for a flapp.

Parameters:

Flapp	The FactoryLink Application Directory that contains the Instance Objects.  Type: String
-------	---

Access: In

Return Value: IObjects

ExportApplicationObject Returns all root Instance Objects for a flapp.

Parameters:

Flapp The FactoryLink Application Directory.

Type: String

Access: In

ClassName Optional, the classname of the object. If this is not specified, the ClassID must be.

Type: String

Access: In

ClassID Optional, the Class ID of the object. If this is not specified, the ClassName must be.

Type: String

Access: In

AccessDatabaseFilePath Optional, the path of the Access Database file to export to. If this is not specified, a file prompt will appear.

The file must either not exist, in which case a blank Access export database will be created, or it must be a valid Access export database.

Type: String

Access: In

Return Value: None

ReturnBlankInstanceFile Returns a path to the file where the blank instance database resides.

ImportApplicationObjects Imports all objects from a valid Access export database into the specified flapp.

Parameters:

Flapp The FactoryLink Application Directory that contains the Instance Object.

Type: String

Access: In

DatabaseFilePath Optional, the path of the Access Database file to import from. If this is not specified, a file prompt will appear.

The file must be a valid Access export database.

Type: String

Access: In

Return Value: None

ModifyIObjectInstance Takes an IObject that has been modified. Each IAOInstance in its IAOInstances property has an IAOTVarInstances object. A specific IAOInstance will be modified. This object has a collection of IAOTVarInstance objects. The newvalue property of each was initially set to Chr\$(0) when the object was retrieved. To modify the object, set the newvalue to the desired value. The ModifyIObject call looks for newvalues that are not chr\$(0). When it finds one, it modifies the Application Object Instance.

The Instance ID of the top most Application Object Instance is returned.

Parameters:

AOIObj The IObject that contains the instance to be modified.

Type: IObject

Access: In/Out

InstanceKey Optional Parameter that is used to find the Instance in the AOIObj. If this is not specified, the InstanceNumber must be.

Type: String

Access: In

InstanceNumber Optional Parameter that is used to find the Instance in the AOIObj. If this is not specified, the InstanceKey must be.

Type: Long

Access: In

	ProvideFeedback	An optional parameter which, if set to true, will cause the feedback events to be fired. The default value is true.
		Type: Boolean
		Access: In
	Return Value: String	
DeleteAllAppObjects		Deletes all the Application Object Classes and Template Variables in the Flapp.
	Parameters:	
	Flapp	The FactoryLink Application Directory.
		Type: String
		Access: In
	Return Value: None	

### **Events:**

Most events processing is done by the event servers, FlinkcdbEvents and AppObjectEvents. The AObjectServer provides VB events relating to instantiation progress. It is necessary to early bind to the AObjectServer object, not IAOServer, to use these events.

BeginFeedBack	Event notifying the user that an action is about to occur.
	Parameters:
	ServerAction
	A Parameter that represents what type of action is about to occur. It is of type AOServerAction that is an enumeration that takes the following values.
	AOSA_Create = 0
	AOSA_Update = 1
	AOSA_Delete = 2
	AOSA_Test = 3
ProvideFeedBack	Event notifying the user that an action has occurred and feedback can be given to the user.
	Parameters:

	Description	A string containing a description of what has occurred.
	Description2	A string containing more detail about what has occurred.
	Cancel	A Boolean which, if set to true by the client application, will cancel operation.
ProvideDetails	Event notifying the user that an action has occurred and feedback can be given to the user. This event provides more detail than providefeedback	
	Parameters:	
	Description	A string containing a description of what has occurred.
EndFeedBack	Event notifying the user that the action has finished	

## IObjects

IObjects is a collection based object containing Objects of type IObject.

Flink7Defns Interface: IAOIObjects

Public: Yes

Creatable: Yes

### **Properties:**

Count	Returns the number of objects in the collection. Type: Long Access: Read
Item	Returns the object based on a key or index. The key is the InstanceID property. Type: IObject Access: Read

### **Methods:**

Add	Creates an IObject and adds it to the collection. Parameters: InstanceID      The InstanceID of the IObject. This is a GUID and therefore is unique. Type: String Access: In/Out InstanceBaseName      The Instance Base Name (Group Base Name) of the IObject. Type: String Access: In/Out Flapp      The Application Directory. Type: String Access: In/Out
-----	---

AppObject	The App Object (IAOObject) that was used to create the instance.
	Type: IAOObject
	Access: In/Out
ParentInstanceID	The InstanceID of the Parent Object.
	Type: String
	Access: In/Out
ParentInstanceNumber	The Instance Number of the Parent Object.
	Type: Long
	Access: In/Out
InstanceDataAvailable	Boolean stating whether the Iobject being added has its IAInstances object filled.
	Type: String
	Access: In

Return Value: Iobject

Remove

Removes an Iobject based on a key from the collection.

Parameters:

vntIndexKey	The key of the object to be removed. The InstanceID is the key.
	Type: Variant
	Access: In/Out



# IObject

IObject is an object, which represents a group of Instantiated Application Objects. Each IObject may contain a group of Instance objects, which represent each instance within the group.

Flink7Defns Interface: IAObject

Public: Yes

Creatable: Yes

## **Properties:**

InstanceID	The Instance ID of the IObject. This is a GUID and therefore is unique. Type: String Access: Read/Write
ParentInstanceID	The Instance ID of the IObject's parent. Type: String Access: Read/Write
ParentInstanceNumber	The Instance Number of the IObject's parent. Type: Long Access: Read/Write
AppObject	The Application Object (IAObject) which was used to create the group of instances. Type: IAObject Access: Read/Write
Flapp	The Application Directory. Type: String Access: Read/Write
InstancebaseName	The Instance Base Name or Group base name of the object. Type: String Access: Read/Write
Instances	The Object of type IAInstances, representing the collection of Instances this IObject has created. It is optional to fill this object with instance data. See the InstanceDataAvailable to determine whether the Instance Data has been loaded.

	Type: IAOInstances
	Access: Read/Write
InstanceDataAvailable	A boolean indicating whether instance data has been loaded into the object.
	Type: String
	Access: Read/Write
GroupDescription	The Group Description of the object.
	Type: String
	Access: Read/Write
InstanceTVarDataAvailable	A boolean indicating whether each instance in the Instances property has had Template Variable data loaded into it. Instance data must have been loaded in order to be able to load Tvar Data.
	Type: Boolean
	Access: Read/Write
OPCItemDataAvailable	A boolean indicating whether each instance in the Instances property has had OPC Item data loaded into it. Instance data must have been loaded in order to be able to load OPC Item Data.
	Type: Boolean
	Access: Read/Write

### **Methods:**

ReturnInstantiatedNodeClass	Returns a NodeClass Key based on an Application Object Type.												
	Parameters:												
	<table> <tr> <td>AOObjectType</td> <td>Optional, the ObjectType to use in the calculation. If this is not specified, the ObjectType of the AppObject property is used.</td> </tr> <tr> <td></td> <td>Type: Enumeration AOAppObjectType</td> </tr> <tr> <td></td> <td>AOFlinkRecord = 0</td> </tr> <tr> <td></td> <td>AOFlinkFile = 1</td> </tr> <tr> <td></td> <td>AOApplicationObject = 2</td> </tr> <tr> <td></td> <td>Access: In/Out</td> </tr> </table>	AOObjectType	Optional, the ObjectType to use in the calculation. If this is not specified, the ObjectType of the AppObject property is used.		Type: Enumeration AOAppObjectType		AOFlinkRecord = 0		AOFlinkFile = 1		AOApplicationObject = 2		Access: In/Out
AOObjectType	Optional, the ObjectType to use in the calculation. If this is not specified, the ObjectType of the AppObject property is used.												
	Type: Enumeration AOAppObjectType												
	AOFlinkRecord = 0												
	AOFlinkFile = 1												
	AOApplicationObject = 2												
	Access: In/Out												
	Return Value: String												

## Instances

Instances is a collection based object containing Objects of type Instance.

Flink7Defns Interface: IAOInstances

Public: Yes

Creatable: Yes

### **Properties:**

Count	Returns the number of objects in the collection.  Type: Long  Access: Read
Item	Returns the object based on the key. The key is the InstanceKey property of the Instance Object.  Type: Instance  Access: Read
ItemByIndex	Returns the object based the Instance Number property of the Instance Object.  Type: Instance  Access: Read

### **Methods:**

Add Creates an Instance object and adds it to the collection.

Parameters:

InstanceKey	The Instancekey of the Instance.  Type: String  Access: In/Out
InstanceNumber	The Instance Number of the Instance.  Type: String  Access: In/Out
InstanceBaseName	The Instance Base Name of the IObject object the Instances collection was created from.

		Type: String
		Access: In/Out
	InstanceDescription	The Description of the Instance.
		Type: String
		Access: In/Out
	InstanceDisabled	A Boolean stating whether the instance has been disabled or not.
		Type: Boolean
		Access: In/Out
	Return Value: Instance	
Remove	Removes an Instance based on a key from the collection.	
	Parameters:	
	vntIndexKey	The key of the object to be removed. The InstanceKey is the key.
		Type: Variant
		Access: In/Out
	Return Value: None	
LastInstanceNumber	Returns the last (highest) Instance Number in the Instances Collection.	
	Return Value: Long	

## Instance

Instance is an object that represents a single instance of an Application Objects.

Flink7Defns Interface: IAOInstance

Public: Yes

Creatable: Yes

### **Properties:**

InstanceNumber	The Instance number of the Instance. Type: Long Access: Read/Write
InstanceBaseName	The Instance Base Name of the group of instances the instance object resides it. Type: String Access: Read/Write
InstanceDescription	The Description of the instance. Type: String Access: Read/Write
InstanceKey	The Key of the instance. Type: String Access: Read/Write
InstanceDisabled	A Boolean stating whether the instance has been disabled. Type: boolean Access: Read/Write
TemplateVariableInstances	A collection based Object containing information about the Instantiated Template Variables associated with the instance. This may not have been initialized. Type: AOTVarInstances Access: Read/Write
NewInstanceDescription	A Property which is initialized to chr\$(0) when the instance is created. Set this property to a new value and invoke the ModifyIObjct method. The new description will be written to the instance.

Type: String

Access: Read/Write

OPCItems

A collection based Object containing information about the OPC Items associated with the instance. This may not have been initialized.

Type: AOOPCItems

Access: Read/Write

### **Methods:**

DisplayName

Returns a string, which is the instance key and, optionally, the instance description. This is obsolete.

Parameters:

UseDescription

Optional, boolean stating whether to include the description property in the display name.

## AOTVarInstances

AOTVarInstances is a collection based object containing Objects of type AOTVarInstance.

Flink7Defns Interface: IAOTVarInstances

Public: Yes

Creatable: Yes

### **Properties:**

Count	Returns the number of objects in the collection.  Type: Long  Access: Read
Item	Returns the object based on the key. The key is the BaseID property of the AOTVarInstance Object.  Type: Instance  Access: Read

### **Methods:**

Add	Creates an AOTvarInstance object and adds it to the collection.  Parameters:						
	<table><tr><td>Name</td><td>The Name of the Template Variable.  Type: String  Access: In/Out</td></tr><tr><td>Value</td><td>The Value of the Template Variable.  Type: String  Access: In/Out</td></tr><tr><td>BaseID</td><td>The ID of the Template variable the AOTVarInstance was derived from.  Type: String  Access: In/Out</td></tr></table>	Name	The Name of the Template Variable.  Type: String  Access: In/Out	Value	The Value of the Template Variable.  Type: String  Access: In/Out	BaseID	The ID of the Template variable the AOTVarInstance was derived from.  Type: String  Access: In/Out
Name	The Name of the Template Variable.  Type: String  Access: In/Out						
Value	The Value of the Template Variable.  Type: String  Access: In/Out						
BaseID	The ID of the Template variable the AOTVarInstance was derived from.  Type: String  Access: In/Out						

Return Value: AOTVarInstance

Remove

Removes an AOTVarInstance based on a key from the collection.

Parameters:

vntIndexKey

The key of the object to be removed. The InstanceKey is the key.

Type: Variant

Access: In/Out

Return Value: None

ReturnValue

Returns a value of a Template Variable based on the key. The key is the base ID or the Template variable ID the AOTVarInstance was derived from.

Return Value: String



## AOTVarInstance

An AOTVarInstance is an object that represents a template variable that was used in an Instance of an Application object.

Flink7Defns Interface: IAOTVarInstance

Public: Yes

Creatable: Yes

### **Properties:**

Value	The Value the Template Variable was assigned. Type: String Access: Read/Write
Name	The Name of the Template Variable. Type: String Access: Read/Write
BaseID	The ID of the Template Variable the Instance was Derived From. Type: String Access: Read/Write
NewValue	When the object is initialized, this property is set to Chr\$(0). When it is set to a different value and the ModifyIObjct method of the AObjectServer is invoked, the new value will be used to modify the configuration. Type: String Access: Read/Write

## AOTVariables

AOTVariables is a collection based object containing Objects of type AOTVariable.

Flink7Defns Interface: IAOTVariables

Public: Yes

Creatable: Yes

### **Properties:**

Count	Returns the number of objects in the collection.  Type: Long  Access: Read
Item	Returns the object based on the key. The key is the ID property of the AOTVariable Object.  Type: AOTVariable  Access: Read

### **Methods:**

Add	Creates a AOTVariable object and adds it to the collection.  Parameters:						
	<table><tr><td>Name</td><td>The Name of the Template Variable.  Type: String  Access: In/Out</td></tr><tr><td>ID</td><td>Optional, the ID of the Template variable. If not specified, a new ID is generated. This is a GUID and therefore is unique. Type: String  Access: In/Out</td></tr><tr><td>Value</td><td>Optional, The Value of the Template Variable.  Type: String  Access: In/Out</td></tr></table>	Name	The Name of the Template Variable.  Type: String  Access: In/Out	ID	Optional, the ID of the Template variable. If not specified, a new ID is generated. This is a GUID and therefore is unique. Type: String  Access: In/Out	Value	Optional, The Value of the Template Variable.  Type: String  Access: In/Out
Name	The Name of the Template Variable.  Type: String  Access: In/Out						
ID	Optional, the ID of the Template variable. If not specified, a new ID is generated. This is a GUID and therefore is unique. Type: String  Access: In/Out						
Value	Optional, The Value of the Template Variable.  Type: String  Access: In/Out						

	DataSource	Optional, a FlinkCDBObjects.cTemplateReplication object (Interface ItemplateReplication) configured with the replication properties.  Type: ItemplateReplication  Access: In/Out
	Presentation	Optional, a FlinkCDBObjects.cTemplatePresentation object (Interface ItemplatePresentation) configured with the Presentation properties.  Type: ItemplatePresentation  Access: In/Out
	Description	The Description of the Template Variable.  Type: String  Access: In/Out
	Return Value:	AOTVariable
Remove		Removes an AOTVariable based on a key from the collection.  Parameters:
	vntIndexKey	The key of the object to be removed. The ID is the key.  Type: Variant  Access: In/Out
	Return Value:	None
TVarByName		Returns an AOTVariable from the collection based on the Template variable Name.  Parameters:
	Name	The name of the variable to find.  Type: String  Access: In
	Return Value:	AOTVariable

## AOTVariable

An AOTVariable is an object that represents a template variable that was used in an Application Object Class.

Flink7Defns Interface: IAOTVariable

Public: Yes

Creatable: Yes

### **Properties:**

Value	The Value of the Template Variable. Type: String Access: Read/Write
Name	The Name of the Template Variable. Type: String Access: Read/Write
ID	The ID of the Template Variable. This is a GUID and therefore is unique. Type: String Access: Read/Write
Presentation	A Presentation object, FlinkCDBObjects.cTemplatePresentation. This contains information on how the template variable is presented to the user. Normally, only prompt variables use it. Type: ITemplatePresentation Access: Read/Write
Replication	A Replication object, FlinkCDBObjects.cTemplateReplication. This contains information on how the template variable receives its values. Type: ITemplateReplication Access: Read/Write
Description	The Description of the Template Variable. Type: String Access: Read/Write

**VariableDataSource** Sets the Variable's Data Source. This property maps is passed directly into the Replication object and is provided so the user does not have to access the Replication object directly.

Type: Enumeration EnumTVDataSource

TVKeyboard (0)  
tvgenerated (1)  
tvTextFile (2)  
tvExcel = (3)  
tvODBC = (4)

Access: Read/Write

**Properties** This is a collection based object of type AOProperties. It holds named value pairs that map into the Replication object and presentation object. The names of the properties it holds depend upon the VariableDataSource and VariablePromptPresentation values. When the Variable Data Source and VariablePromptPresentation is set, the object is filled with the pertinent properties from the two objects necessary to describe the variable. These can be modified and are saved when the Update Method is invoked. This object was provided to so that the user does not have to access the Replication object and presentation object directly

Type: AOProperties

Access: Read/Write

**VariablePromptPresentation** Sets the Variable's Presentation Style. This property maps is passed directly into the Presentation object and is provided so the user does not have to access the Presentation object directly.

Type: Enumeration EnumAOPresentationStyles

AOPTextBox (0)  
AOPComboBox (1)  
AOPListBox (2)  
AOPCheckBox (3)  
AOPRadio (4)

Access: Read/Write

**Flapp** The Application Directory.

Type: String

Access: Read/Write

### **Methods:**

**ValidName** Checks a name for validity and returns true if it is a valid name. This does not check to see if the variable name already exists.

Parameters:

Name	The Name of the Template Variable to check.
	Type: String
	Access: In
ErrorMessage	Optional, If included, an error message with the reason the name is invalid is filled in
	Type: String
	Access: Out

Return Value: Boolean

Update

Updates a template variable in the database using the Properties collection. Use the AddVariableToDatabase method on the AObjectServer object when using the Replication and Presentation Objects directly.

Parameters:

AObjectServer	Optional, an AObjectServer variable. If not included, a new AObjectServer object is created.
	Type: AObjectServer
	Access: In/Out

Return Value: None

Delete

Deletes this template variable from the database.

Parameters:

AObjectServer	Optional, an AObjectServer variable. If not included, a new AObjectServer object is created.
	Type: AObjectServer
	Access: In/Out

Return Value: None

## AOProperties

AOProperties is a collection based object containing Objects of type AOProperty.

Flink7Defns Interface: IAOProperties

Public: Yes

Creatable: Yes

### **Properties:**

Count	Returns the number of objects in the collection.  Type: Long  Access: Read
Item	Returns the object based on the key. The key is the Name property of the AOProperty Object.  Type: AOProperty  Access: Read

### **Methods:**

Add	Creates an AOProperty object and adds it to the collection.  Parameters: <table><tr><td>Name</td><td>The Name of the Property. The name of the property is the Key  Type: String  Access: In/Out</td></tr><tr><td>Value</td><td>The value of the Template variable.  Type: String  Access: In/Out</td></tr></table>  Return Value: AOProperty	Name	The Name of the Property. The name of the property is the Key  Type: String  Access: In/Out	Value	The value of the Template variable.  Type: String  Access: In/Out
Name	The Name of the Property. The name of the property is the Key  Type: String  Access: In/Out				
Value	The value of the Template variable.  Type: String  Access: In/Out				
Remove	Removes an AOProperty based on a key from the collection.  Parameters:				

vntIndexKey

The key of the object to be removed. The Name is the key.

Type: Variant

Access: In/Out

Return Value: None



## AOProperty

An AOProperty is an object that represents a general named-value pair. The Name property is given the name of the property and the value is given the value.

Flink7Defns Interface: IAOProperty

Public: Yes

Creatable: Yes

### **Properties:**

Value	The Value of the Property. Type: String Access: Read/Write
Name	The Name of the Property. Type: String Access: Read/Write

## AOOPCItems

AOOPCItems is a collection based object containing Objects of type AOOPCItem.

Flink7Defns Interface: IAOOPCItems

Public: Yes

Creatable: No

### **Properties:**

Count	Returns the number of objects in the collection.  Type: Long  Access: Read
Item	Returns the object based on the key. The key is the Name property of the AOOPCItem Object.  Type: AOOPCItem  Access: Read
MiscProperties	This is a collection based object of type AOProperties. It holds named value pairs. Currently the property holds the RecordPath and the RowID of the FactoryLink Record that the config object created.  Type: AOProperties  Access: Read/Write

### **Methods:**

Add	Creates an AOOPCItem object and adds it to the collection.  Parameters:				
	<table><tr><td>ItemName</td><td>The OPC Item Name. In FactoryLink, this is the Tag Name. This is the Key for the Collection.  Type: String  Access: In</td></tr><tr><td>ItemPropertyName</td><td>The Name of the configuration field which resulted in this OPC Item. In FactoryLink, this is the name of the field in the AC file.</td></tr></table>	ItemName	The OPC Item Name. In FactoryLink, this is the Tag Name. This is the Key for the Collection.  Type: String  Access: In	ItemPropertyName	The Name of the configuration field which resulted in this OPC Item. In FactoryLink, this is the name of the field in the AC file.
ItemName	The OPC Item Name. In FactoryLink, this is the Tag Name. This is the Key for the Collection.  Type: String  Access: In				
ItemPropertyName	The Name of the configuration field which resulted in this OPC Item. In FactoryLink, this is the name of the field in the AC file.				

Type: String

Access: In

ItemPropertyDescription The Description of the configuration field which resulted in this OPC Item. In FactoryLink, this is the Description of the field in the AC file.

Type: String

Access: In

ItemDefinition The Template Variable, if any, which resulted in this OPC Item.

Type: String

Access: In

ItemType The OPC Item Type. Currently only items of type FlinkTag(0) are valid.

Type: OPCItemType

Access: In

Return Value: AOOPCItem

Remove

Removes an AOOPCItem based on a key from the collection. The Key is the OPC Item.

Parameters:

vntIndexKey The key of the object to be removed. The Name is the key.

Type: Variant

Access: In/Out

Return Value: None

## AOOPCItem

An AOOPCItem is an object that represents an OPCItem created indirectly through Application Objects.

Flink7Defns Interface: IAOOPCItem

Public: Yes

Creatable: No

### **Properties:**

**ItemPropertyDescription** The Description of the configuration field which resulted in this OPC Item. In FactoryLink, this is the Description of the field in the AC file.

Type: String

Access: Read/Write

**ItemPropertyName** The Name of the configuration field which resulted in this OPC Item. In FactoryLink, this is the name of the field in the AC file.

Type: String

Access: Read/Write

**ItemName** The OPC Item Name. In FactoryLink, this is the Tag Name.

Type: String

Access: Read/Write

**ItemType** The OPC Item Type. Currently only items of type FlinkTag(0) are valid.

Type: OPCItemType

Access: Read/Write

**MiscProperties** This is a collection based object of type AOProperties. It holds named value pairs. Currently this property is not used.

Type: AOProperties

Access: Read/Write

**ItemDefinition** The Template Variable, if any, which resulted in this OPC Item.

Type: String

Access: Read/Write

## AObjects

AObjects is a collection based object containing Objects of type AObject.

Flink7Defns Interface: IAObjects

Public: Yes

Creatable: Yes

### **Properties:**

Count	Returns the number of objects in the collection. Type: Long Access: Read
Item	Returns the object based on the key. The key is the Name property of the AObject Object. Type: AProperty Access: Read

### **Methods:**

Add	Adds an existing AObject to the collection. Parameters: AO                      The AObject to be added. The Key is the AObject ID. Type: AObject Access: In/Out Return Value: AObject
Remove	Removes an AObject based on a key from the collection. Parameters: vntIndexKey            The key of the object to be removed. The AObject ID is the key. Type: Variant Access: In/Out Return Value: None

## AObject

An AObject is an object that represents an Application Object class.

Flink7Defns Interface: IAObject

Public: Yes

Creatable: Yes

### **Properties:**

ID	<p>The ID of the Application Object. The ID is a GUID.</p> <p>Type: String</p> <p>Access: Read/Write</p>
ClassName	<p>The Name Object Class. This should be unique across all object classes in the system.</p> <p>Type: String</p> <p>Access: Read/Write</p>
Description	<p>The description of the class.</p> <p>Type: String</p> <p>Access: Read/Write</p>
Variables	<p>The Tvariables the object uses. This is an object of type AOTVariables</p> <p>Type: AOTVariables</p> <p>Access: Read/Write</p>
ObjectType	<p>The Type of the object. Each Object type has it's own set of unique properties in the properties collection and performs it's own set of actions. Once an object has been created and saved, changing its object type is not supported.</p> <p>Type: Enumeration AOAppObjectType</p> <ul style="list-style-type: none"><li>AOFlinkRecord (0)</li><li>AOFlinkFile (1)</li><li>AOApplicationObject (2)</li></ul> <p>Access: Read/Write</p>
DefaultInstances	<p>The default number of instances the object should create when it is first run. This parameter is overridden if the AObjectServer method call contains a</p>

Number of Instances parameter, although this will apply only to the top level object affected by the call. Sub objects will continue to use their DefaultInstances property when they are created. This must be a string that can be converted to a long integer, a template variable or a ?.

Type: String

Access: Read/Write

## Properties

This is a collection based object of type AOProperties. It holds named value pairs. The names of the properties it holds depend upon the ObjectType. When the Object is first created, usually with the Create method on the AObjectServer Object, the properties collection is filled with properties pertinent to that type of object. These can be modified and are saved when the Update Method on the AObjectServer Object is invoked.

Currently, the following properties are supported

Object Type

AOFlinkRecord (0):

ParentTablePath	The path of the Parent table, if there is one. It is in the form task/Domain/table/table... For example, al_def/SHARED/al_group/al_info  Type: String  Access: In/Out
ParentTableFields	The field values that each parent table needs to have set. It is in the form table='Value', table='Value'. For example, al_group='GRPNAM', al_info='NUM'. The field values can be template variables.  Type: String  Access: In/Out
RecordTablePath	The path of the table to be written to. It is in the form task/Domain/table/table... For example, al_def/SHARED/al_group/al_info/al_rel  Type: String  Access: In/Out
RecordTableFields	The field values need to be written into the application. It is in the form Field1='Value', Field2='Value2' For example, ATAG='{TagName}',COND='ON',MSG='{

AlarmDescription}',PRIO='1'. The field values can be template variables.

Type: String

Access: In/Out

**RecordFieldAttributes** The field attributes that values need to be written into the application. Currently, only FactoryLink tags have attributes. It is in the form Field1=AttrString1, Field2=AttrString2. For example, ATAG=SHARED"|1" { AlarmDescription} "| ||False|False|False|False| The AttrString is a packed string representing Tag Attributes. This string can encoded or decoded by using the AttributeString property on the FlinkCDBObjects.cTagAttributes object.

Type: String

Access: In/Out

**DuplicateRecordAction** The action to take when a record with the same values in the fields specified in DuplicateRecordFields already exists.

Type: Enumeration

AODuplicateRecordAction  
AOAllowDuplicates = 0  
AOLeaveAsIs = 1  
AOUpdate = 2

Access: In/Out

**DuplicateRecordFields** The fields to examine for matching values to determine whether there is a duplicate record. Format is Field1,Field2. For example, ATag, AID

Type: String

Access: In/Out

**AOFlinkFile (1):**

**FileRecordPath** The path of the file to create. It is in the form:  
task/Domain/table!\$FILENAME=  
filename.ext,

for example:

IMLP/SHARED/IMLTAGS!\$FILENAME=  
dplogsim.prg



	This can property can have template variables in it.
	Type: String
	Access: In/Out
FileExistsAction	<p>What to do when the file exists. The choices are to replace the file or to leave it alone. If the FileObjectCreation property is AOFOCSameFile (0), this value must be AOFileReplace (1)</p> <p>Type: Enumeration AOFileExistsAction  AOFileLeave (0)  AOFileReplace (1)</p> <p>Access: In/Out</p>
FileObjectCreation	<p>Where to place each File Object instance. Valid choices are in the same file, or in separate files.</p> <p>Type: Enumeration AOFileObjCreation  AOFOCSameFile (0)  AOFOCSeparateFiles (1)</p> <p>Access: In/Out</p>
FileHeaderText	<p>The text to place in the beginning of each file. If each instance is in separate files, this text is written into each file. If each instance is in the same file, this text is written once at the beginning. This can property can have template variables in it.</p> <p>Type: String</p> <p>Access: In/Out</p>
InstanceText	<p>The text to place in the file each time an instance is created. This can property can have template variables in it.</p> <p>Type: String</p> <p>Access: In/Out</p>
FileFooterText	<p>The text to place at the end of each file. If each instance is in separate files, this text is written into each file. If each instance is in the same file, this text is written once at the end. This can property can have template variables in it.</p>

Type: String

Access: In/Out

AOApplicationObject (2):

Currently, objects of type AOApplicationObject do not have custom properties filled into their properties collection.

InstanceBaseName	<p>The base name of the group of instances. This property can take a template variable. If the GroupBaseName exists, the instance numbers will be assigned, starting at the end of the last instance, otherwise, the instance numbers will be assigned starting at 1.</p> <p>Type: String</p> <p>Access: Read/Write</p>
InstanceDescription	<p>The Description of the Instance. This property can take template variables.</p> <p>Type: String</p> <p>Access: Read/Write</p>
InstanceKey	<p>The key of the Instance. If this property is specified, it must be a template variable that resolves to a unique key for the object.</p> <p>Type: String</p> <p>Access: Read/Write</p>
InstanceDisableFlagField	<p>This field is obsolete.</p> <p>Type: String</p> <p>Access: Read/Write</p>
GroupDescription	<p>The Description of the Base Group. This property can take template variables.</p> <p>Type: String</p> <p>Access: Read/Write</p>
ClassClosedIcon	<p>An optional string that can specify a custom closed icon for an Application Object class in the FlinkTree. The icon must already exist in the tree or the custom icon must have been added to the tree 16x16 icons directory. When adding an icon or bitmap to the directory, it is referred to by the filename without the extension (case sensitive). The current editors do not have the ability to specify icons. Note that the update method must be called on the AObjectServer or on the AObject to save the icon. This property may have template variables in it.</p> <p>Type: String</p> <p>Access: Read/Write</p>

ClassOpenedIcon	<p>An optional string that can specify a custom opened icon for an Application Object class in the FlinkTree. See ClassClosed Icon Property.</p> <p>Type: String</p> <p>Access: Read/Write</p>
InstanceClosedIcon	<p>An optional string that can specify a custom closed icon for an Application Object Group of Instances in the FlinkTree. See ClassClosed Icon Property.</p> <p>Type: String</p> <p>Access: Read/Write</p>
InstanceOpenedIcon	<p>An optional string that can specify a custom opened icon for an Application Object Group of Instances in the FlinkTree. See ClassClosed Icon Property.</p> <p>Type: String</p> <p>Access: Read/Write</p>
ParameterString	<p>Reserved for future use.</p> <p>Type: String</p> <p>Access: Read/Write</p>
HTMLPropertyPage	<p>An optional string that specifies a user defined WebPage to use to display properties. The page should exist in the {Flapp}\AppObj directory.</p> <p>Type: String</p> <p>Access: Read/Write</p>
DisableVariable	<p>An template variable (including the { }) that Application Objects check to see if it matches the DisableValue to determine whether to disable the object or not.</p> <p>Type: String</p> <p>Access: Read/Write</p>
DisableValue	<p>An value that Application Objects check to see if it matches the value of the DisableVariable value to determine whether to disable the object or not.</p> <p>Type: String</p> <p>Access: Read/Write</p>
IAddressOffset	<p>Used with the IaddressMultiplier to skip rows in the datasources. The row accessed is given by the below. Note that the Block Offset properties in the editors do not show these values. Instead they calculate the values they display.</p> $\text{AddressIndex} = (\text{IAddressMultiplier} * \text{InstanceNumber}) + \text{IAddressOffset}$

Type: Long

Access: Read/Write

IAddressMultiplier

Used with the IAddressOffset to skip rows in the datasources. The row accessed is given by the below. Note that the Block Offset properties in the editors do not show these values. Instead they calculate the values they display.

$$\text{AddressIndex} = (\text{IAddressMultiplier} * \text{InstanceNumber}) + \text{IAddressOffset}$$

Type: Long

Access: Read/Write

Member

Optional qualifier used to differentiate between multiple classes of the same type. When the object is added to the AObjects, if there is an identical object, the member is used to differentiate between them.

Type: String

Access: Read/Write

Flapp

The Application Directory.

Type: String

Access: Read/Write

### **Methods:**

FindVariable

Retrieves a template variable based upon its name. The name is not case sensitive.

Parameters:

Name

The name of the Template variable.

Type: String

Access: In

Return Value: AOTVariable

TvarsUsedInConfig

Returns a variant array containing the template variables used in the object's configuration. The array is dimensioned from 0 to n, where n is the number of variables. The 0<sup>th</sup> dimension is ignored.

Parameters:

Name

The name of the Template variable.

Type: String

Access: In

	Return Value: Variant	
IsContainer	Returns true if the Object can function as a container. Currently only objects of type AOApplicationObject (2) can be containers.	
	Return Value: Boolean	
DisplayName	Returns a concatenated string that can be used as an object description.	
	Parameters:	
	UseDescription	Optional, boolean. If false, the description is the classname, the member, enclosed in parentheses, if the member is a non-empty string, and the table if the object is of type AOFlinkRecord. If true, the description is as above, but with the Description appended to it.
		Type: Boolean
		Access: In
	UseBulletChar	Optional, boolean. If false, when the description is separated from the rest of the string by a -, otherwise it is separated from the rest of the string by CHR\$(149) (The bullet when using the Tahoma font).
		Type: Boolean
		Access: In
	Return Value: String	
ReturnAOCClass	Returns an string representing the object class.	
	Parameters:	
	AObjectType	Optional parameter. If missing, the Object type property of the AOObject is used, otherwise it is used.
		Type: variant
		Access: In
	Return Value: String	
ValidName	Checks a name for validity and returns true if it is a valid name. This does not check to see if the variable name already exists.	
	Parameters:	

Name	The Name of the Object to check.
	Type: String
	Access: In
ErrorMessage	Optional, If included, an error message with the reason the name is invalid is filled in
	Type: String
	Access: Out

Return Value: Boolean

#### VariablesAsList

Returns a variant array containing the template variables used in the object. This differs from TvarsUsedInConfig in that the list consists of all variables added to the object, not just those used in configuration. The array is dimensioned from 0 to n, where n is the number of variables. The 0<sup>th</sup> dimension is ignored.

Return Value: Variant

## AOIObjData

The AOIObjData object is a special object available for Instances of Configuration of file objects. It contains the information related to the configuration written by the object.

Flink7Defns Interface: IAIOObjData

Public: Yes

Creatable: Yes

### **Properties:**

InstanceID	The Instance ID of the IObject that created the configuration. Type: String Access: Read/Write
InstanceNumber	The Instance number of the Instance that created the configuration. Type: Long Access: Read/Write
ObjectType	The type of object the data pertains to. Type: Enumeration AOAppObjectType AOFlinkRecord (0) AOFlinkFile (1) AOApplicationObject (2) Access: Read/Write
CFGObjTablePath	The Table path used to write the configuration record. Only valid if the ObjectType was AOFlinkRecord. Type: String Access: Read/Write
CFGObjRowid	The RowID of the configuration record. Only valid if the ObjectType was AOFlinkRecord. Type: String Access: Read/Write
Table	The short name of the table of the configuration record. Only valid if the ObjectType was AOFlinkRecord.

	Type: String
	Access: Read/Write
Flapp	The Application Directory.
	Type: String
	Access: Read/Write
FilePath	The path of the File that was created by the Object. Only valid if the ObjectType was AOFileLinkFile.
	Type: String
	Access: Read/Write
RecordPath	The path of the File that was created by the Object in cRecordPath format. Only valid if the ObjectType was AOFileLinkFile.
	Type: String
	Access: Read/Write
OwnedByAO	The owner of the File that the Object attached to. Only valid if the ObjectType was AOFileLinkFile.
	Type: Enumeration AOFileOwner
	AOFileOwnedByAO (0)
	AOFileOwnedByOther (1)
	Access: Read/Write