Cumulus 5 OLE Automation Overview

For information on using Cumulus Automation Objects in applications supporting OLE Automation, please refer to the file "Cumulus.tlb", which is located in the application’s directory.   Below you will find a list of the various currently supported objects.

Cumulus 4 backward compatibility

The following list describes how to implement Cumulus 4 OLE automation methods and properties using the new Cumulus 5 OLE automation interface.

Cumulus 4 (removed methods, properties and objects)	**Cumulus 5 equivalents**
AssetReference.**Text**	AssetReference.**Path**
AssetReference.**Type**	AssetReference.**Name**
Catalog.**CollectionFolder**	Catalog.CentralAssetLocation
Catalog.**CollectionFolderUsage**	Catalog.CentralAssetLocationUsage
Catalogs Object	Use the CatalogCollections and Application objects instead.
CatalogingOptions.**Categories**	This property can be controlled using the field linking and initial value method. Take the definition object as shown in CatalogingOptions.**Status**, but use UID_REC_CATEGORIES instead: Set Def = defs.ItemByID(UID_REC_CATEGORIES) Def.LinkWhenAdding = False ' ignore data from the cataloger Def.UseInitialValue = True Def.InitialValue = "My Category" If you want to set multiple categories, you can concatenate them like this: Def.InitialValue = "A" & Chr$(13) & Chr$(10) & "B" & Chr$(13) & Chr$(10) & "C"
CatalogingOptions **CopyAssetToCollectionFolder**	CatalogingOptions.CopyAssetToCentralLocation
CatalogingOptions **Notes** and **IncludeNotesFromAsset**	The Cumulus 4 options CatalogingOptions.IncludeNotesFromAsset = False CatalogingOptions.Notes = "Hello World" can be controlled using the field linking and initial value method. Take the definition object as shown in CatalogingOptions.**Status**, but use UID_REC_STATUS instead: ... Set Def = defs.ItemByID(UID_REC_STATUS) Def.LinkWhenAdding = False ' ignore data from the cataloger Def.UseInitialValue = True Def.InitialValue = "My secret notes !"
CatalogingOptions.**Status**	This property can be controlled using the field linking and initial value method.

```
Dim defs As FieldDefinitions
Dim Def As FieldDefinition
Dim L As Layout

' Take a Collection from anywhere...
Set L = Collection.Catalog.RecordLayout
Set defs = L.FieldDefinitions
Set Def = defs.ItemByID(UID_REC_STATUS)

Def.LinkWhenAdding = False ' ignore data from the cataloger
Def.UseInitialValue = True
Def.InitialValue = "Status X"
```

The text (here "Status X") this must be a valid entry of the UID_REC_STATUS value list (Def.ValueList)

CatalogingOptions.**User**

This propertz can be controlled in the same way as shown above in CatalogingOptions.**Status**, but use the UID_REC_USER field instead:

```
...
Set Def = defs.ItemByID(UID_REC_USER)
Def.LinkWhenAdding = False ' ignore data from the cataloger
Def.UseInitialValue = True
Def.InitialValue = "Mr. X"
```

CatalogingOptions.
IncludeCategoriesFromAsset

This property can be controlled by switching the field linking the category field off / on. Take the definition object as shown in CatalogingOptions.**Status**, but use UID_REC_CATEGORIES instead:

```
...
Set Def = defs.ItemByID(UID_REC_CATEGORIES)
Def.LinkWhenAdding = False ' no categories will be created.
```

Category.**Directory** — Category().ItemByID(UID_CAT_WIN_FOLDER_PATH)

Category.**Name** — Category().ItemByID(UID_CAT_NAME)

Category.**Type** — Category().ItemByID(UID_CAT_TYPE)

CategoryTypeConstants
NormalCategory, RelatedCategory, DirectoryCategory

Use the five new constants now:
NormalCategory, RelatedCategory, DirectoryCategoryMac, DirectoryCategoryWin, DirectoryCategoryURL

CollectionFolderUsageConstants — CentralAssetLocationUsageConstants

FileFormatTable Object	AssetFormatTable Object
FileFormatTable.**Add**	Not available
FileFormatTable.**Remove**	Not available
FileFormatEntry Object	AssetFormatEntry Object
FileFormatEntry.**DefaultBitmap**	Not available
LogEntries Object	Removed, no equivalence available
LogEntry Object	Removed, no equivalence available
Record.**AssetCreationDate**	Record().ItemByID(UID_REC_ASSET_CREATION_DATE)
Record.**AssetDataSize**	Record().ItemByID(UID_REC_ASSET_DATA_SIZE)
Record.**AssetModificationDate**	Record().ItemByID(UID_REC_ASSET_MODIFICATION_DATE)
Record.**AssetRsrcSize**	Record().ItemByID(UID_REC_ASSET_RSRC_SIZE)
Record.**AssetSize**	' This is the sum of Record().ItemByID(UID_REC_ASSET_DATA_SIZE) + Record().ItemByID(UID_REC_ASSET_RSRC_SIZE)
Record.**ColorMode**	Record().ItemByID(UID_REC_COLOR_MODE)
Record.**CopyThumbnail**	Record().ItemByID(UID_REC_THUMBNAIL).CopyPicture Record.Save
Record.**Deletable**	Record().ItemByID(UID_REC_ASSET_TAG_CANNOT_BE_DELETED) ' Note: this is the inverse value of Record.Deleteable!
Record.**Deselect**	CatalogCollection.SelectedRecords.Remove Record
Record.**FileFormat**	Record().ItemByID(UID_REC_FORMAT)
Record.**Height**	Record().ItemByID(UID_REC_HEIGHT)
Record.**MacCreator**	Record().ItemByID(UID_REC_MAC_FILE_CREATOR)
Record.**MacFileType**	Record().ItemByID(UID_REC_MAC_FILE_TYPE)
Record.**Modifiable**	Record().ItemByID(UID_REC_ASSET_TAG_CANNOT_BE_MODIFIED) ' Note: this is the inverse value of Record.Modifiable!
Record.**Name**	Record().ItemByID(UID_REC_RECORD_NAME)

Record.**Notes**	Record().ItemByID(UID_REC_NOTES)
Record.**OpenPropertyWindow**	Record.ShowInformationWindow
Record.**PasteThumbnail**	Record().ItemByID(UID_REC_THUMBNAIL).PastePicture Record.Save
Record.**RecordCreationDate**	Record().ItemByID(UID_REC_ASSET_TAG_CREATION_DATE)
Record.**RecordLayout**	CatalogCollection.Catalog.RecordLayout
Record.**RecordModificationDate**	Record().ItemByID(UID_REC_ASSET_TAG_MODIFICATION_DATE)
Record.**Select**	CatalogCollection.SelectedRecords.Add Record
Record.**Software**	Record().ItemByID(UID_REC_CREATING_SOFTWARE)

Record.**Status**.Name	Dim ID As Integer
	ID = Record().ItemByID(UID_REC_STATUS)
	MsgBox
	collection.catalog.RecordLayout().ItemByID(UID_REC_STATUS).ValueList.ItemByID(ID)

Record.**TypeSpecific**	Record().ItemByID(UID_REC_FORMAT_SPECIFIC) Instead of handling a value as a hex string, you have to get/set the raw binary data. Please check the example.
Record.**User**	Record().ItemByID(UID_REC_USER)
Record.**Width**	Record().ItemByID(UID_REC_WIDTH)
SelectedCategories Object	Use the Categories object instead.
SelectedRecords Object	Use the Records object instead.
StatusList / **Status** Object	Dim Status As String Dim StatusID As Integer Dim StatusList As ValueList Set StatusList = collection.catalog.RecordLayout().ItemByID(UID_REC_STATUS).ValueList Status = StatusList(0) StatusID = StatusList(0).ID
SubCategories Object	Use the Categories object instead.

UnitConstants Centimeter, Inch, Millimeter, Pica, Pixel, Point	Use the prefix "u": uCentimeter, uInch, uMillimeter, uPica, uPixel, uPoint

See also Overview

Name As String

This property returns the name of the application.

This is the **default** property.

See also Application

Fullname As String

This property returns the full path of the application's executable.

See also Application

Application As Application

This property returns the Application object.

See also Application

GenericEvent (ByVal EventID As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal n3 As Long, ByVal n4 As Long)

If you have connected this event handler to Cumulus, you will be notified after certain actions.

The master argument EventID contains the ID of the event. The arguments n1, n2 [, n3, n4] specify the affected objects of this event. The following table shows all available events that are fired from Cumulus.

The values in brackets represent the internal constant values (EV_RECORD_ADDED = 100).

Catalog related events

EventID	**n1**	**n2**
EV_RECORD_ADDED (100) A record has been added to the catalog (cataloged).	CatalogID	RecordID
EV_RECORD_UPDATED (101) A record in the catalog has been updated (modified).	CatalogID	RecordID
EV_RECORD_DELETED (102) A record has been deleted from the catalog.	CatalogID	RecordID
EV_CATEGORY_ADDED (110) A category has been added to the catalog (cataloged).	CatalogID	CategoryID
EV_CATEGORY_UPDATED (111) A category in the catalog has been updated (modified).	CatalogID	CategoryID
EV_CATEGORY_DELETED (112) A category has been deleted from the catalog.	CatalogID	CategoryID
EV_RECORD_FIELD_ADDED (160) A new record field has been added to the Record Layout.	CatalogID	ShortID
EV_RECORD_FIELD_UPDATED (161) A record field has been updated (modified) in the Record Layout.	CatalogID	ShortID
EV_RECORD_FIELD_DELETED (162) A record field has been deleted from the Record Layout.	CatalogID	ShortID
EV_CATEGORY_FIELD_ADDED (165)	CatalogID	ShortID

A new category field has been added to the Category Layout.		
EV_CATEGORY_FIELD_UPDATED (166) A category field has been updated (modified) in the Category Layout.	CatalogID	ShortID
EV_CATEGORY_FIELD_DELETED (167) A category field has been deleted from the Category Layout.	CatalogID	ShortID
EV_CATALOG_CLOSED (150) A catalog has been closed.	CatalogID	0
EV_CATALOG_PROPS_CHANGED (151) The catalog properties (catalog name, thumbnail size etc.) have been changed.	CatalogID	0

General collection related UI events

EventID	**n1**	**n2**
EV_UI_COLLECTION_CLOSED (351) The Collection (a catalog window) has been closed.	CollectionID	0
EV_UI_RECORD_REMOVED (300) The user has removed the record from the Collection (not from the catalog).	CollectionID	RecordID
EV_UI_RECORD_DELETED (301) The user (not another client) has removed the record from the Collection AND deleted the record from the catalog.	CollectionID	RecordID
EV_UI_CATEGORY_DELETED (310) The user (not another client)has removed the category from the Collection AND deleted the category from the catalog.	CollectionID	CategoryID
EV_UI_NEW_ROOT_CATEGORY (311) The user (not another client) has created a new root category.	CollectionID	CategoryID

Mouse related UI events

EventID	n1	n2
EV_UI_RECORD_L_DBL_CLICK (321) The user has left double-clicked a record.	CollectionID	CollectionIndex
EV_UI_CATEGORY_L_DBL_CLICK (331) The user has left double-clicked a category.	CollectionID	CategoryID

Selection related UI events

EventID	n1	n2
EV_UI_RECORD_SELECTION_CHANGED (400) The user has changed the selection of records.	CollectionID	0
EV_UI_CATEGORY_SELECTION_CHANGED (401) The user has changed the selection of categories.	CollectionID	0

Miscellaneous global events

EventID	n1	n2
EV_UI_EXIT (350) The Application (Cumulus) has been terminated.	0	0
EV_UI_APP_PREF_CHANGED (600) The Application Preferences have been changed. The argument n1 describes which part of the preferences have been modified (AppPrefChangedEventConstants). The argument n2 contains the new value of this preference. In case of What = AppPrefUnit the New Value is one of the UnitConstants values.	What	New Value

If you want to catch events, you have to declare your application object in a separate class (for example "MyClass") using the keyword "WithEvents". Your main form has to contain an object of the type MyClass.

VB sample code:

```
******************** Form FireTest.frm ********************
   Dim A As New MyClass

   Private Sub Form_Load()
      A.init
   End Sub
```

```
******************** Class MyClass.cls ********************

    Public WithEvents MyApp As Cumulus5.Application

    ' init creates the automation object
    Sub init()
        Set MyApp = CreateObject("cumulus5.application")
        MyApp.Visible = True
    End Sub

    ' Event Handler, chosen from the MyClass event list
    Private Sub MyApp_GenericEvent(ByVal EventID As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal n3 As Long, ByVal n4
    As Long)
        If (EventID = EV_CATALOG_CLOSED) Then
            MsgBox "EV_CATALOG_CLOSED: " & n1
        End If
    '------------------
        If (EventID = EV_UI_COLLECTION_CLOSED) Then
            MsgBox "EV_UI_COLLECTION_CLOSED: " & n1
        End If
    '------------------
        If (EventID = EV_CATEGORY_DELETED) Then
            MsgBox "EV_CATEGORY_DELETED: " & n1 & " " & n2
        End If
    '------------------
        If (EventID = EV_RECORD_FIELD_ADDED) Then
            MsgBox "EV_RECORD_FIELD_ADDED: " & n1 & " " & n2
        End If

    End Sub
```

See also Application

CatalogingOptions As CatalogingOptions

This property returns the Cataloging Options object.

See also Application

Parent As Object

This property returns the application’s Application object as parent.

See also Application

UsedApplicationList As Variant

This read-only property returns the list of the most recently used applications (only the name, not the path of the executable) for opening (editing) or printing an asset. This list does not necessarily contain the asset's associated program. Use Record.AssociatedProgram to   this.

These values can be used for the ApplicationName argument for Record.PrintTheAsset or Record.EditTheAsset.

VB sample code:

```
Dim i As Integer
Dim size As Long
Dim Apps() As String

Apps = Application.UsedApplicationList
size = UBound(Apps) - LBound(Apps) + 1 ' get the size

For i = 0 To size - 1
    MsgBox Apps(i)
Next i
```

See also Application

ConvertToUTF16 (UTF8 As String) As String

This property converts a UTF-8 encoded string to the Windows native code (treated as UTF-16). ConvertToUTF16 can be used to decode the contents of a UTF-8 encoded Cumulus export file (Record or Category).

The reverse property is ConvertToUTF8.

VB sample code:

```
Dim utf8 As String
utf8 = "xÃ“Ã–ÃœxÃ¤Ã¶Ã¼Ã©Ã¨" ' this is a UTF-8 string
MsgBox Application.ConvertToUTF16(utf8)
```

This will display "xÄÖÜxäöüéè"

See also Application

ConvertToUTF8 (UTF16 As String) As String

This property converts a Windows native string (treated as UTF-16) to the UTF-8 character encoding. This property can be used to create a UTF-8 encoded Cumulus export file (Record or Category).

The reverse property is ConvertToUTF16.

VB sample code:

```
Dim utf16 As String
utf16 = "xÄÖÜxäöüéè" ' this is a Windows native string
MsgBox Application.ConvertToUTF8(utf16)
```

This will display "xÃ“Ã–ÃœxÃ¤Ã¶Ã¼Ã©Ã¨"

See also Application

FindToken (TokenID As Long) As String

The property returns a token string that can be used to create a language-independent find query string. The argument TokenID specifies the wanted token.

In the following sample, the find request using the QueryEnglish query string works only on a Cumulus installation with an English user interface.

The second query string is language-independent because the FindToken property returns the localized string. The localized field name is requested using the FieldDefinition Object.

VB sample code:

```
Dim coll As CatalogCollection
Dim Query As String
Dim QueryEnglish As String
Dim Name As String

Set coll = colls.FrontCollection
Name = coll.Catalog.RecordLayout.FieldDefinitions.ItemByID(UID_REC_RECORD_NAME).Name

QueryEnglish = "Record Name" & vbTab & "starts with" & vbTab & "A"
Query = Name & vbTab & a.FindToken(FindStartsWith) & vbTab & "A"

MsgBox coll.Find(QueryEnglish, FindNew)
MsgBox coll.Find(Query, FindNew)
```

Possible values for TokenID are:

FindContains	= 0
FindStartsWith	= 1
FindIs	= 2
FindIsNot	= 3
FindDoesntContain	= 4
FindUnder	= 5
FindUpTo	= 6
FindFrom	= 7
FindOver	= 8
FindPrior	= 9
FindAfter	= 10
FindThumbSimilar	= 11
FindThumbNotSimilar	= 12
FindHasNoValue	= 13
FindHasValue	= 14
FindBoolTrue	= 15
FindBoolFalse	= 16
FindAnd	= 17
FindOr	= 18
FindCategoriesField	= 19

See also Application, FindTokenConstants

Preferences As Preferences

This property returns the application's Preferences object.

See also Application

PaletteMode As Boolean

This property returns or sets the application’s palette view state.

See also Application

Visible As Boolean

This property returns or sets the application’s visibility state. If the application is invisible, the application window and program icon in the taskbar of Windows disappear.

See also Application, Activate

Activate

This method brings the Cumulus application window to the front. If the application was previously in the background, minimized or invisible, this state will be changed to an activated foreground window.

See also Application, Visible

DeleteAsset (AssetReferenceXML As String)

This method deletes the asset specified by its XML asset reference data.

See also Application

CopyAsset (SrcAssetReferenceXML As String, DstAssetReferenceXML As String) As String

This method creates a copy of the source asset specified by its XML asset reference data. The new asset is created within the given destination asset. The XML asset reference of the asset copy is returned after successfully copying the asset.

See also Application

CountContainedAssets (AssetReferenceXML As String) As Long

This method returns the number of assets contained in the asset specified by its XML asset reference.

See also Application

GetContainedAsset (AssetReferenceXML As String, IndexOrName As String / Long) As String

This method returns the XML asset reference of an asset contained in another asset (of the contained asset). The container asset is specified by its XML asset reference. The contained asset is specified either by index or name.

See also Application

ListContainedAssetNames (AssetReferenceXML As String) As Variant

This method returns a list of asset names (array of string) for all assets that are contained in the given container asset.

See also Application

ConnectToServer (ServerName As String, User As String, Password As String) As Server

This method connects to the specified server as the given user using the password and returns the Server object on success.

If User and Password are left blank, a "Guest" connection is established.

```
Dim S As Server
Set S = app.ConnectToServer("Monster", "", "")
```

If you want to deviate from the last TCP Port Number used, you can add a special Server Port Number, separated by a colon:

```
Set S = app.ConnectToServer("Monster:32456", "John", "abcdefg")
```

See also Application

NewCatalog (Pathname As String [, DatabaseSelectorUID As String]) As CatalogCollection

This method creates a new catalog file, opens it and returns the CatalogCollection object.

The argument DatabaseSelectorUID (a GUID string value) allows you to select a specific database engine. If you omit this argument or set it to "", the Cumulus Engine is used as default. Some possible values are:

UID_SELECTOR_CUE (Cumulus Engine)
UID_SELECTOR_ODBC (ODBC Connector)

It is allowed to specify the name of the database selector instead of the UID (for example "Cumulus Database Engine" or "ODBC Database Connector").

Note: Not available for Cumulus Clients.

See also Application, FieldAccessConstants

OpenCatalog (Pathname As String [, DatabaseSelectorUID As String]) As CatalogCollection

This method allows you to open an existing catalog file and returns the CatalogCollection object.

The argument DatabaseSelectorUID (a GUID string value) allows you to select a specific database engine. If you omit this argument or set it to "", the Cumulus Engine is used as default. Some possible values are:

UID_SELECTOR_CUE (Cumulus Engine)
UID_SELECTOR_ODBC (ODBC Connector)

It is allowed to specify the name of the database selector instead of the UID (for example "Cumulus Database Engine" or "ODBC Database Connector").

Note: Not available for Cumulus Clients.

See also Application, FieldAccessConstants

Quit

This method quits the running Cumulus application.

See also Application

RestoreFactoryDefaults

This method restores the original (factory) catalog layout (fields and properties) as default values for the creation of new catalogs. If you create a new catalog, it will have this layout. See also Catalog.SetAsDefault.

See also Application

OpenCatalogByReference (CatalogReference As String, bWithoutRecords As Long) As CatalogCollection

This method opens a catalog using its reference data and returns the CatalogCollection object. A dialog may appear asking to enter a user password to connect to a server or to enter a password for write access.

The CatalogReference can be taken from the Catalog.CatalogReference property.

The parameter bWithoutRecords allows to create the CatalogCollection window without any visible records (if set to 1).

See also Application

LanguageID As Long

This read-only property returns the language that is used for the Cumulus application.

The value is one of the language constants and can be used to parameterize the methods and properties of the Layout object.

See also Application, LanguageConstants

RestoreCatalog (BackupPathName As String, PathName As String) As CatalogCollection

This method restores a catalog backup file (that was previously created with Catalog.Backup), opens the created catalog and returns the CatalogCollection object.

See also Application

FrontCollection As CatalogCollection

This property returns the application’s frontmost CatalogCollection object.

See also Application

CatalogCollections As CatalogCollections

This property returns the application's CatalogCollections object. This is the list of all open collection windows.

See also Application

Add (ModuleClassUID As String, FieldUID As String [, AsIndex As Long]) As LinkEntry

The behavior of this method depends on the origin of the LinkList object.

FieldDefinition.LinkToFields:
This methods adds a new LinkEntry to the LinkList and returns the created LinkEntry object. The arguments ModuleClassUID and FieldUID define the module and its field as link targets for the field described by the FieldDefinition object. The arguments can be taken from the Layout.AvailableFields LinkList.

You can specify an optional AsIndex value to determine the position of the new entry. If you do not specify the AsIndex value, the new field as appended at the end of the LinkList.

Layout.AvailableFields:
This method is not available if the LinkEntry object is taken from Layout.AvailableFields.

VB sample code:

```
Dim AF As LinkList
Dim LL As LinkList
Dim e As LinkEntry
Dim eNew As LinkEntry
Set AF = Layout.AvailableFields
Set LL = FieldDefinition.LinkToFields
Set e = AF(2)
MsgBox LL.Count
Set eNew = LL.Add(e.ModuleClassUID, e.FieldUID, 4)
MsgBox LL.Count ' the number of entries is increased by one
```

See also LinkList

AddAllMatchingModuleFields

The behavior of this method depends on the origin of the LinkList object.

FieldDefinition.LinkToFields:
This methods adds new LinkEntries to the LinkList. All entries from the Layout.AvailableFields LinkList with the same FieldUID or the same name as the field described by the FieldDefinition object will be added.

Layout.AvailableFields:
This method is not available if the LinkEntry object is taken from Layout.AvailableFields.

See also LinkList

Application As Application

This property returns the Application object.

See also LinkList

Count As Long

This property returns the number of link entries in the LinkList collection.

See also LinkList

Item (Index As Long) As LinkEntry

This method gets a specific LinkEntry object from the LinkList collection. Item can be used by Index (of the collection).

This is the **default** property.

See also LinkList

MoveLinkEntry (FromIndex As Long, ToIndex As Long)

This method is not available any longer.

See also LinkList

Parent As Object

This property returns the LinkList’s parent object.

See also LinkList

Remove (Index As Long)

The behavior of this method depends on the origin of the LinkList object.

FieldDefinition.LinkToFields:
This method removes a specific LinkEntry object from the LinkList collection. Remove can be used by Index (of the collection).

Layout.AvailableFields:
This method is not available if the LinkEntry object is taken from Layout.AvailableFields.

See also LinkList

Application As Application

This property returns the Application object.

See also LinkEntry

FieldUID As String

This property returns the FieldUID (a GUID string value) of this link entry.

The properties ModuleClassUID and FieldUID define the module and its field as link targets.

See also LinkEntry, FieldAccessConstants

Name([LanguageID As Long]) As String

This read-only property returns the name of the field represented by the link entry's FieldUID.

If you do not specify a language ID, the current Application.LanguageID value is taken.

This is the **default** property.

See also LinkEntry

ModuleClassUID As String

This property returns the ModuleClassUID (a GUID string value) of this link entry.

The properties ModuleClassUID and FieldUID define the module and its field as link targets.

See also LinkEntry, FieldAccessConstants

ModuleName([LanguageID As Long]) As String

This read-only property returns the name of the module represented by the link entry's ModuleClassUID.

If you do not specify a language ID, the current Application.LanguageID value is taken.

See also LinkEntry

Property FieldValueList([LanguageID]) As ValueList

This read-only property returns the ValueList object belonging to this field. A field has a ValueList if the field type is FieldTypeEnum.

See also LinkEntry

Parent As Object

This property returns the LinkEntry’s parent object.

See also LinkEntry

FieldType As Long

This property returns the type of this field (represented by the LinkEntry object).

The different field type constants are listed in LayoutFieldTypeConstants.

Possible values are:

FieldTypeString	= 0
FieldTypeBool	= 1
FieldTypeInteger	= 2
FieldTypeDouble	= 3
FieldTypeDate	= 4
FieldTypeBinary	= 5
FieldTypePicture	= 6
FieldTypeEnum	= 7
FieldTypeAudio	= 8

See also LinkEntry

Application As Application

This property returns the Application object.

See also Groups

Parent As Object

This property returns the Groups' parent object.

See also Groups

Count As Long

This property returns the number of group entries in the Groups collection.

See also Groups

Remove (Index As Long / Name As String)

This method removes a specific Group object from the Groups collection. Remove can be used by Index (of the collection) or by Name (of the group).

If any users are assigned to the removed group (see UserPermission.GroupID), they are reassigned to the "Others" group.

The "Others" group (Index = 0) can't be removed.

See also Groups

RemoveByID (ID As Long)

This method removes a specific Group object (specified by the ID) from the Groups collection.

If any users are assigned to the removed group (see UserPermission.GroupID), they are reassigned to the "Others" group.

The "Others" group (Index = 0) can't be removed.

See also Groups

Item (Index As Long / Name As String) As Group

This property returns a Group object by index or by name.

This is the **default** property.

See also Groups

ItemByID (ID As Long) As Group

This property returns a Group object by group ID.

See also Groups

Add (Name As String) As Group

This method adds a new group to the user Groups collection using the given name and returns the new created group as Group object.

See also Groups

Application As Application

This property returns the Application object.

See also Group

Parent As Object

This property returns the Group’s parent object.

See also Group

ID As Long

This read-only property returns the group’s internal ID. You can assign a user to this group by setting its Group ID (see UserPermission.GroupID) to the desired group.

See also Group

Name As String

This property returns or sets the group’s name.

This is the **default** property.

See also Group

Application As Application

This property returns the Application object.

See also AssetStorage

Parent As Object

This property returns the AssetStorage's parent object.

See also AssetStorage

Active As Boolean

This property returns or sets if this asset storage entry is active.

See also AssetStorage

AssetStorageUID As String

This read-only property returns the ID of this asset storage (a GUID string value like UID_AS_WIN_FILE, UID_AS_MAC_FILE, UID_AS_URL).

See also AssetStorage, FieldAccessConstants

Name ([LanguageID As Long])

This read-only property returns the name of this asset storage. If you do not specify a language ID, the current Application.LanguageID value is taken.

This is the **default** property.

See also AssetStorage, LanguageConstants

Application As Application

This property returns the Application object.

See also AssetStorages

Parent As Object

This property returns the AssetStorages’ parent object.

See also AssetStorages

Count As Long

This property returns the number of asset storage entries in the AssetStorages collection.

See also AssetStorages

Item (Index As Long) As AssetStorage

This method gets a specific AssetStorage object from the AssetStorages collection. Item can be used by Index (of the collection).

This is the **default** property.

See also AssetStorages

ItemByID (AssetStorageUID As String) As AssetStorage

This method gets a specific AssetStorage object from the AssetStorages collection. Item is used by the AssetStorageUID (a GUID string value like UID_AS_WIN_FILE, UID_AS_MAC_FILE, UID_AS_URL).

See also AssetStorages, FieldAccessConstants

Add (ShortID As Long / FieldUID As String / Name As String[, AsIndex As Long]) As ViewEntry

This method adds a new displayed record field to the view and returns the new ViewEntry object. The field must be a valid field from the RecordLayout and can be specified as ShortID (a Long value) or by FieldUID (a GUID string value) or the field name. You can specify an optional AsIndex value to determine the position of the new entry. If you do not specify the AsIndex value, the new field as appended at the end of the list of displayed record fields.

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
The method inserts a new column in the catalog collection's details view and adds this entry to the ViewConfiguration collection.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
The method inserts a new field to each record in the catalog collection's thumbnail view and adds this entry to the ViewConfiguration collection.

VB sample code:

```
    Dim Details As ViewConfiguration
    Dim E As ViewEntry
    Set Details = Collection.DetailsViewConfiguration
    Set E = Details.Add(UID_REC_ASSET_MODIFICATION_DATE, 2)
```

See also ViewConfiguration

Application As Application

This property returns the Application object.

See also ViewConfiguration

BackgroundColor As Long

This property returns or sets the background color of the view.

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
The property gets / sets the background color of the catalog collection's details view.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
The property gets / sets the background color of the catalog collection's thumbnails view.

VB sample code:

```
Dim Thumbnails As ViewConfiguration
Set Thumbnails = Collection.ThumbnailViewConfiguration
Thumbnails.BackgroundColor = RGB(255, 255, 20)
```

See also ViewConfiguration

Count As Long

This property returns the number of ViewEntry entries in the ViewConfiguration collection (shown fields in the view).

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
Count returns the number of columns in the catalog collection's details view.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
Count returns the number of shown fields of each record in the catalog collection's thumbnail view.

See also ViewConfiguration

Item (Index As Long / Name As String) As ViewEntry

This method gets a specific ViewEntry object from the ViewConfiguration collection. Item can be used by Index (of the collection) or by Name (of the Field).

This is the **default** property.

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
The range of index / name covers the complete set of shown fields in the catalog collection's details view.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
The range of index / name covers the complete set of shown fields in the catalog collection's thumbnail view.

VB sample code:

```
Dim Details As ViewConfiguration
Set Details = Collection.DetailsViewConfiguration
MsgBox Details.Item(2)
MsgBox Details.Item("File Format")
```

See also ViewConfiguration

ItemByID(ShortID As Long / FieldUID As String) As ViewEntry

This method gets a specific ViewEntry object from the ViewConfiguration collection. ItemByID can be used by ShortID (a Long value) or by FieldUID (a GUID string value).

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
The range of valid IDs covers the complete set of shown fields (columns) in the catalog collection's details view.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
The range of valid IDs covers the complete set of shown fields in the catalog collection's thumbnail view.

VB sample code:

```
Dim Thumbnails As ViewConfiguration
Dim ShortID As Long

Set Thumbnails = Collection.ThumbnailViewConfiguration
ShortID = Collection.Catalog.RecordLayout.FieldDefinitions.ItemByID(UID_REC_FORMAT).ShortID

MsgBox Thumbnails.ItemByID(UID_REC_FORMAT)
MsgBox Thumbnails.ItemByID(ShortID)
```

See also ViewConfiguration

MoveViewEntry (FromIndex As Long, ToIndex As Long)

This methods moves the specified ViewEntry object (FromIndex) within the ViewConfiguration collection. The target position is specified by the ToIndex argument.

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
The method moves a column in the catalog collection's details view to the target column position.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
The method moves a field of each record in the catalog collection's thumbnail view to the target position.

```
Dim Thumbnails As ViewConfiguration
Set Thumbnails = Collection.ThumbnailViewConfiguration
Thumbnails.MoveViewEntry 4, 0
```

See also ViewConfiguration

Parent As Object

This property returns the ViewConfiguration's parent object.

See also ViewConfiguration

Remove (Index As Long / Name As String)

This method removes a specific ViewEntry object from the ViewConfiguration collection. Remove can be used by Index (of the collection) or by Name (of the Field).

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
The range of index / name covers the complete set of shown fields in the catalog collection's details view. The specified column is removed from the details view.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
The range of index / name covers the complete set of shown fields in the catalog collection's thumbnail view. The specified field is removed from the thumbnail view.

VB sample code:

```
Dim Details As ViewConfiguration
Set Details = Collection.DetailsViewConfiguration
Details.Remove(5)
Details.Remove("File Format")
```

See also ViewConfiguration

RemoveByID(ShortID As Long / FieldUID As String)

This method removes a specific ViewEntry object from the ViewConfiguration collection. RemoveByID can be used by ShortID (a Long value) or by FieldUID (a GUID string value).

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
The range of valid IDs covers the complete set of shown fields (columns) in the catalog collection's details view. The specified column is removed from the details view.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
The range of valid IDs covers the complete set of shown fields in the catalog collection's thumbnail view. The specified field is removed from the thumbnail view.

VB sample code:

```
Dim Thumbnails As ViewConfiguration
Dim ShortID As Long

Set Thumbnails = Collection.ThumbnailViewConfiguration
ShortID = Collection.Catalog.RecordLayout.FieldDefinitions.ItemByID(UID_REC_FORMAT).ShortID

Thumbnails.RemoveByID(UID_REC_CATEGORIES)
Thumbnails.RemoveByID(ShortID)
```

See also ViewConfiguration

RestoreDefaults

This method reinitializes the ViewConfiguration collection object and sets it to the values that were previously stored with SaveAsDefaults in the system registry.

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
The previous state of the catalog collection's details view is restored (order of shown fields (columns), fonts, background color, etc.).

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
The previous state of the catalog collection's thumbnail view is restored (order of shown fields, fonts, background color, thumbnail spacing, thumbnail image size, etc.).

See also ViewConfiguration

SaveAsDefaults

This method saves the complete current state of the ViewConfiguration collection (list of ViewEntry objects containing font attributes, background color, etc.) permanently to the system registry.

The default values can be restored later using the RestoreDefaults method.

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
The current state of the catalog collection's details view is saved as defaults (order of shown fields (columns), fonts, background color, etc.).

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
The current state of the catalog collection's thumbnail view is saved as defaults (order of shown fields, fonts, background color, thumbnail spacing, thumbnail image size, etc.).

See also ViewConfiguration

ShowFrame As Boolean

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
This method is not available if the ViewConfiguration object is taken from CatalogCollection.DetailsViewConfiguration.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
This property returns or sets if the thumbnails are decorated with a 3D frame in the catalog collection's thumbnail view.

See also ViewConfiguration

ThumbnailImageSize As Long

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
This method is not available if the ViewConfiguration object is taken from CatalogCollection.DetailsViewConfiguration.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
This property returns or sets the thumbnail image size value. This is the pixel size of shown thumbnails in the catalog collection's thumbnail view. The allowed range is from 16 to 1024.

See also ViewConfiguration

ThumbnailSpacing As Long

The behavior of this method depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
This method is not available if the ViewConfiguration object is taken from CatalogCollection.DetailsViewConfiguration.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
This property returns or sets the thumbnails spacing value. This is the additional pixel spacing around the area of shown thumbnails in the catalog collection's thumbnail view. The allowed range is from 0 to 1024.

See also ViewConfiguration

Application As Application

This property returns the Application object.

See also ViewEntry

Bold As Boolean

This property returns or sets the **Bold** state of the font for this view entry.

See also ViewEntry

Color As Long

This property returns or sets the Color of the font for this view entry.

VB sample code:

```
Dim Details As ViewConfiguration
Dim E As ViewEntry
Set Details = Collection.DetailsViewConfiguration
Set E = Details.ItemByID(UID_REC_RECORD_NAME)
E.Color = RGB(255, 54, 190)
```

See also ViewEntry

ColumnWidth As Long

The behavior of this property depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
This property returns or sets the column width (in pixels) of this view entry in the catalog collection's details view.

CatalogCollection.ThumbnailViewConfiguration:
CatalogCollection.PaletteViewConfiguration:
This method is not available if the ViewEntry object is taken from CatalogCollection.ThumbnailViewConfiguration / CatalogCollection.PaletteViewConfiguration:

See also ViewEntry

CharSet As Long

This property returns or sets the CharSet state of the font for this view entry. The value can be one of the CharSetConstants and can be used to define the language character set of the font for this view entry.

See also ViewEntry, CharSetConstants

FieldUID As String

This property returns the FieldUID (a GUID string value) of this view entry.

You can use the FieldUID to request a FieldDefinition object to obtain more information about the field properties in the catalog.

VB sample code:

```
Dim Def As FieldDefinition
Set Def = Collection.Catalog.RecordLayout.FieldDefinitions.ItemByID(ViewEntry.FieldUID)
```

See also ViewEntry

FontName As String

This property returns or sets the name of the font for this view entry.

VB sample code:

```
Dim Details As ViewConfiguration
Dim E As ViewEntry
Set Details = Collection.DetailsViewConfiguration
Set E = Details.ItemByID(UID_REC_RECORD_NAME)
E.FontName = "Times New Roman"
```

See also ViewEntry

FontSize As Long

This property returns or sets the size (in point) of the font for this view entry.

See also ViewEntry

Italic As Boolean

This property returns or sets the *Italic* state of the font for this view entry.

See also ViewEntry

Name([LanguageID As Long]) As String

This read-only property returns the name of the field represented by the view entry.

If you do not specify a language ID, the current Application.LanguageID value is taken.

This is the **default** property.

The value can also be obtained by requesting the FieldDefinition object (see also ViewEntry.FieldUID).

VB sample code:

```
Dim E As ViewEntry
Set E = Collection.DetailsViewConfiguration.ItemByID(UID_REC_CATEGORIES)
MsgBox E.Name
' will display the same as
MsgBox Collection.Catalog.RecordLayout.FieldDefinitions.ItemByID(E.FieldUID).Name
```

See also ViewEntry, LanguageConstants

Parent As Object

This property returns the ViewEntry's parent object.

See also ViewEntry

StrikeOut As Boolean

This property returns or sets the ~~StrikeOut~~ state of the font for this view entry.

See also ViewEntry

Underline As Boolean

This property returns or sets the Underline state of the font for this view entry.

See also ViewEntry

Application As Application

This property returns the Application object.

See also Fields

ListOfData(What As Long) As Variant

This read-only property returns a list of information taken from all (!) Field entries. You can use this function to get the information faster and more efficiently than calling the Fields.Item function in a loop to request single Field objects and call their properties. The returned array contains the information in the same order as if you would loop over the single items by index. The array corresponds to the list returned from FieldDefinitions.ListOfData.

The "What" argument specifies what kind of information is returned and can have one of the following values:

ListOfFieldType = 1
Returns an array of Long (equivalent to FieldDefinition.FieldType).

ListOfFieldUID = 2
Returns an array of String (equivalent to Field.FieldUID).

ListOfHasValue = 5
Returns an array of Boolean (equivalent to Field.HasValue).

ListOfValue = 6
Returns an array of Variant (equivalent to Field.Value).
There are some special cases:

In case of the field type **FieldTypePicture,** the Variant value of the array entry is "Empty". In this case, you have to use Field.Value or CopyPicture / CopyPictureToFile.

In case of the field type **FieldTypeBinary** or **FieldTypeAudio,** the Variant value of the array entry is itself an array of Byte representing the binary data (see sample code).

In case of the virtual record field **UID_REC_CATEGORIES**, the Variant value of the array entry is itself an array of Long representing the IDs of categories assigned to this record (see sample code).

VB sample code:

```
Dim fs As Fields
Dim count As Long
Dim i As Long
Dim j As Long
Dim size As Long

Dim Values As Variant
Dim HasValues As Variant
Dim IDs As Variant
Dim Types As Variant

Set fs = Record.Fields ' Record taken from anywhere
count = fs.count

Values = fs.ListOfData(ListOfValue)
IDs = fs.ListOfData(ListOfFieldUID)
HasValues = fs.ListOfData(ListOfHasValue)
Types = fs.ListOfData(ListOfFieldType)


For i = 0 To count - 1
    ' special case categories
    If (IDs(i) = UID_REC_CATEGORIES) Then
```

```
            Dim cx() As Long
            cx = Values(i)
            ' get the size, might be zero if no categories are assigned to the record
            size = UBound(cx) - LBound(cx) + 1
            For j = 0 To size - 1
                MsgBox cx(j) ' shows the category id
            Next
        Else
            If Types(i) = FieldTypeBinary Or Types(i) = FieldTypeAudio Then ' binary case
                If HasValues(i) Then
                    Dim b() As Byte
                    b = Values(i) ' get the data
                    size = UBound(b) - LBound(b) + 1     ' get the size
                    MsgBox "Binary Field, size = " & size
                Else
                    MsgBox "Binary Field, no value"
                End If
            Else ' regular case
                MsgBox IDs(i) & Chr(13) & HasValues(i) & Chr(13) & Values(i) & Chr(13) & Types(i)
            End If
        End If
    Next
```

See also Fields, ListOfDataConstants

Parent As Object

This property returns the Fields’ parent object.

See also Fields

Count As Long

This property returns the number of field entries in the Fields collection.

See also Fields

Item (Index As Long / Name As String) As Field

This method gets a specific Field object from the Fields collection. Item can be used by Index (of the collection) or by Name (of the Field).

This is the **default** property.

Note: ItemByID used with the FieldUID (GUID string) is the fastest way of access. Item (Name) is slower.

VB sample code (please see the Value documentation for more sample code):

```
Dim fs As Fields
Set fs = Record.Fields ' We want to change a record

fs.Item("Record Name").Value = "Hello Record"
MsgBox fs.Item("Record Name").Value ' will display "Hello Record"

' Using the default properties:
Record("Record Name") = "ABC"
MsgBox Record("Record Name") ' will display "ABC"
MsgBox fs(13) ' will display the value (content) of the 13th field

Record.Save ' needed to make the changes permanent

Set fs = Category.Fields ' now we want to change a category

fs.Item("Category Name").Value = "Hello Dolly"
fs("Category Name") = "Hello Category" ' overwrites "Hello Dolly"
MsgBox fs.Item("Category Name") ' will display "Hello Category"

' There is no need to call a Category.Save because the changing of category values takes effect immediately.
```

The different field names of a Record or a Category can be requested using the Catalog.RecordLayout or Catalog.CategoryLayout.

See also Fields

ItemByID (ShortID As Long / FieldUID As String) As Field

This method gets a specific Field object from the Fields collection. ItemByID can be used by ShortID (a Long value) or by FieldUID (a GUID string value).

ItemByID used with the FieldUID (GUID string) is the fastest way of access (faster than Item).

VB sample code (please see the Value documentation for more sample code):

```
Dim fs As Fields
Set fs = Record.Fields ' We want to change a record

fs.ItemByID(UID_REC_HEIGHT).Value = 1234 ' Request uses the GUID string
MsgBox fs.ItemByID(UID_REC_HEIGHT).Value ' will display 1234

Record.Save ' needed to make the changes permanent

' Using the default properties:
MsgBox fs.ItemByID(UID_REC_HEIGHT) ' will also display 1234
MsgBox fs.ItemByID(1214605671) ' will display the value of the field with the given ShortID

Set fs = Category.Fields ' now we want to change a category

fs.ItemByID(UID_CAT_NAME).Value = "Hello Dolly"
fs.ItemByID(UID_CAT_NAME) = "Hello Category" ' overwrites "Hello Dolly"
MsgBox fs.ItemByID(UID_CAT_NAME) ' will display "Hello Category"

' There is no need to call a Category.Save because the changing of category values takes effect immediately.
```

The different predefined field id constants are listed in FieldAccessConstants. The different FieldUIDs and ShortIDs of a Record or a Category can be requested using the Catalog.RecordLayout or Catalog.CategoryLayout.

See also Fields

Application As Application

This property returns the Application object.

See also Field

HasValue As Boolean

This property returns if this field has a value or clears the value of the field.

After creating a new field in the Catalog.RecordLayout or Catalog.CategoryLayout, the field will be initially empty (has no value) in all Records and Categories of the catalog. After this field is assigned a value, it has a value.

You can clear the value of a field by assigning false to the HasValue property. It is NOT possible to assign HasValue = true.

VB sample code:

```
' assuming we have created a new Integer field named "MyIntField"
Dim r As Record
Dim fs As Fields
Dim f As Field

' get r from somewhere...
Set fs = r.Fields
Set f = r.Fields("MyIntField")

MsgBox f.HasValue 'returns false
f = 123
MsgBox f.HasValue 'returns true
HasValue = False ' clears the field again (does NOT set to zero)
r.Save
```

After **clearing** fields of a **record** object, you have to call the Record.Save method to clear the values permanently in the catalog file AND to see an updated Record Collection View (catalog window).
Note: It is not allowed to clear the virtual category field UID_REC_CATEGORIES. Use the Record.Categories object instead.

After **clearing** fields of a **category** object, you do NOT have to call a Save method to clear the values permanently in the catalog file. The values are cleared directly.
Note: It is not allowed to clear the following basic category fields:

UID_CAT_NAME
UID_CAT_WIN_FOLDER_PATH
UID_CAT_URL_FOLDER_PATH
UID_CAT_MAC_VOLUME_ID
UID_CAT_MAC_DIRECTORY_ID
UID_CAT_TYPE

See also Field

Parent As Object

This property returns the Field’s parent object.

See also Field

FieldUID As String

This property returns the FieldUID (as a GUID string) of this field.

The FieldUID is a GUID in string representation like "{5251a742-727c-11d2-a73d-0000c000cdd3}".

See also Field

Value As Variant

This property returns or sets the value of a field. The return (or set) value type depends on the type of the field (LayoutFieldTypeConstants). You can use the Catalog.RecordLayout or Catalog.CategoryLayout object to have access to the field types.

This is the **default** property. Note: When reading binary values, you may not omit the Value property (see example below).

The return value of the get Value property is "Empty" if the field was not previously set. You can also check this with the HasValue property.

Do not use this Value property to get / set fields of the field type "FieldTypePicture". Use the methods CopyPicture / CopyPictureToFile (for get) or PastePicture / PastePictureFromFile (for set) instead.

Do not use this Value property to get / set the virtual UID_REC_CATEGORIES field. Use the Record.Categories object instead.

After **setting** fields of a **record** object, you have to call the Record.Save method to store the values permanently in the catalog file AND to see an updated Record Collection View (collection window).

After **setting** fields of a **category** object, you do NOT have to call a Save method to store the values permanently in the catalog file. The values are stored directly.

VB sample code setting / getting string data (FieldTypeString)

```
Record.Fields.ItemByID(UID_REC_RECORD_NAME).Value = "My New Name"
' or shorter:
Record().ItemByID(UID_REC_RECORD_NAME) = "My New Name"

MsgBox Record().ItemByID(UID_REC_RECORD_NAME)
'Important to save the changes permanently after setting all fields:
Record.Save
```

For more samples, see Fields.Item and Fields.ItemByID.

The most efficient way to access fields is given if you use FID Values. These Values can be either be taken from the predefined FieldAccessConstants or can be requested from the layout object as Catalog.RecordLayout.FieldDefinitions.Item(n).FieldUID.

The most inefficient way to access fields is given if you use the field name to select a field and should not be used if you access in loops.

VB sample code for setting / getting binary data (FieldTypeBinary):

```
Private Sub Test_Binary(r As Record)
    Dim s1 As String
    Dim b() As Byte
    Dim size As Integer
    Dim i As Integer

    s1 = "Hi, this is binary data"
    ' convert the string to binary data
    Dim data(22) As Byte ' index 0..22, size = 23
    For i = 0 To 22
        data(i) = Asc(Mid(s1, i + 1, 1))
    Next
```

```
        'instead of r.Fields("bin").Value = data
        r("bin") = data ' set the Data
        b = r("bin").Value ' get the Data, the default property Value does not work here

        size = UBound(b) - LBound(b) + 1 ' get the size
        MsgBox size ' show the size of the binary data, 23 will appear

        ' convert the binary data to a string
        s1 = ""
        For i = LBound(b) To UBound(b)
            s1 = s1 & Chr$(b(i))
        Next
        MsgBox s1 ' "Hi, this is binary data" will appear

        'Important to save the changes permanently after setting all fields:
        r.Save

    End Sub
```

VB sample code for setting / getting binary data (FieldTypeBinary):

VB sample code setting / getting enumeration string data (FieldTypeEnum)

If you read / write field values of a **FieldTypeEnum** field, you always handle with **IDs** of the corresponding Value List. (see FieldDefinition.ValueList, ValueList object).

Fields of the type FieldTypeEnum never store the index, but always the ID of an entry in the corresponding Value List. The following sample demonstrates how to get the string value of the "Status" field of a record.

```
    Private Sub Test_StatusList(coll As CatalogCollection)
        Dim Status As String
        Dim StatusID As Integer
        Dim StatusList As ValueList
        Dim r As Record
        Dim field As field
        Dim ID As Long
        Dim ListEntryID As Long
        Dim ListEntryValue As String


        'Get Status field of a record
        Set r = coll.Records(0)
        Set field = r.Fields.ItemByID(UID_REC_STATUS)

        'get the ID-value of the status field (ID of the entry in the value list)
        ID = field.Value

        'Get the List of Possible Values from the RecordLayout.
        Set StatusList = coll.Catalog.RecordLayout.FieldDefinitions.ItemByID(UID_REC_STATUS).ValueList

        ' Now we get the string value from the list be field value as ID
        ListEntryValue = StatusList.ItemByID(ID).Value
        ListEntryID = StatusList.ItemByID(ID).ID
        MsgBox ListEntryValue
        MsgBox ListEntryID ' has the same value as ID
        ' Now we assign a new value:
        field.Value = StatusList.Item(1).ID

        ' display the ID of the entry with the zero based index 1 of the Status Value List
        MsgBox field.Value

        'Important to save the changes permanently after setting all fields:
        r.Save

    End Sub
```

See also Field, LayoutFieldTypeConstants

CopyPicture

This method copies a picture field to the clipboard.

Use this method instead of reading the value with the Value property in case of FieldType = FieldTypePicture.

See also Field, LayoutFieldTypeConstants

CopyPictureToFile (FileName As String)

This method copies a picture field to the specified file (Windows Bitmap file format, *.bmp). If the file already exists, it will be overwritten.

Use this method instead of reading the value with the Value property in case of FieldType = FieldTypePicture.

See also Field, LayoutFieldTypeConstants

PastePicture

This method inserts a picture field with a bitmap from the clipboard.

Use this method instead of setting the value with the Value property in case of FieldType = FieldTypePicture.

After setting fields of a **record** object, you have to call the Record.Save method to store the values permanently in the catalog. In case of a **category** object, this is not necessary.

VB sample code:

```
Record().ItemByID(UID_REC_THUMBNAIL).PastePicture
Record.Save

Category("MyPicture").PastePicture
' Save not necessary
```

See also Field, LayoutFieldTypeConstants

PastePictureFromFile (FileName As String)

This method replaces a picture field with a bitmap from the specified file (Windows Bitmap file format, *.bmp).

Use this method instead of setting the value with the Value property in case of FieldType = FieldTypePicture.

After setting fields of a **record** object, you have to call the Record.Save method to store the values permanently in the catalog. In case of a **category** object, this is not necessary.

VB sample code:

```
Record().ItemByID(UID_REC_THUMBNAIL).PastePictureFromFile "C:\MyThumbnail.bmp"
Record.Save

Category("MyPicture").PastePictureFromFile "C:\MyThumbnail.bmp"
' Save not necessary
```

See also Field, LayoutFieldTypeConstants

Remove (Index As Long / Name As String)

This methods removes the Record specified by index or by name from the catalog window (collection). This record is not deleted from the catalog, but removed from the catalog view.

The record is removed from the last search result.

See also CatalogCollection

RemoveCategory (Index As Long / Name As String) (Enterprise Edition only)

This methods removes the Category specified by index or by name from the category window (collection). This category is not deleted from the catalog, but removed from the category view.

The category is removed from the last search result.

See also CatalogCollection

ID As Long

This property returns the catalog collection's unique ID. This ID is valid until the collection window is closed. If the same collection is reopened, it will have another ID.

See also CatalogCollection

FindMatchingRecords (CategoryIDs As Variant)

This method finds all records that have at least ALL (!) of the specified categories. The search result is the new content of the collection.

The argument of FindMatchingRecords can be a single CategoryID value (long) or a list of CategoryIDs (array of long).

VB sample code:

```
Dim Recs As Records

Dim List(1) As Long
List(0) = 5
List(1) = 6

' coll is a CatalogCollection object taken from anywhere

' Find all records that have at least the category with ID = 5
coll.FindMatchingRecords 5

' Find all records that have at least the categories with ID = 5 and 6
coll.FindMatchingRecords List
```

See also CatalogCollection

RemoveByID (ID As Long)

This methods removes the Record specified by the ID from the catalog window (collection). This record is not deleted from the catalog, but removed from the collection.

The record is removed from the last search result.

See also CatalogCollection

RemoveCategoryByID (ID As Long) (Enterprise Edition only)

This methods removes the Category specified by the ID from the category window (collection). This category is not deleted from the catalog, but removed from the collection.

The category is removed from the last search result.

See also CatalogCollection

Add (Record As Record)

This methods adds the given Record Object to the collection. This record must be an existing record in the catalog. The record object can be taken from another view of the same catalog that contains it.

The record is added to the last search result.

See also CatalogCollection

AddByID (ID As Long)

This methods adds the Record specified by the ID to the collection (records collection). The ID must refer to an existing record in the catalog

The record is added to the last search result.

See also CatalogCollection

MoveRecord (Index As Long / Name As String, AsIndex As Long)

This methods moves the Record position specified by index or by name within the collection. The target position is specified by the AsIndex argument.

The record changes the display order of the last search result.

The current Sorting state will be cleared and is undefined after moving a record.

See also CatalogCollection

MoveRecordByID (ID As Long, AsIndex As Long)

This methods moves the Record position specified by the ID within the collection. The target position is specified by the AsIndex argument.

The record changes the display order of the last search result.

The current Sorting state will be cleared and is undefined after moving a record.

See also CatalogCollection

Sorting As (Index As Long / Name As String)

This property returns or sets the sorting of the current collection.

If you **set** the Sorting, you can specify either the Index (of the layout) or the Name (of the Field).

```
CatalogCollection.Sorting = 3
CatalogCollection.Sorting = "Record Name"
```

If you **get** the Sorting, you always get the Name (of the field) as String.

```
Dim s As String
s = CatalogCollection.Sorting ' returns "Record Name"
```

In case of an invalid sorting (for example, after moving single records to other positions using MoveRecord), the return string is empty.

See also CatalogCollection

SortingByID As (ShortID As Long / FieldUID As String)

This property returns or sets the sorting of the current collection.

If you **set** the Sorting, you can specify either the ShortID (a Long value) or the FieldUID (a GUID string value).

```
CatalogCollection.Sorting = UID_REC_COLOR_MODE
CatalogCollection.Sorting = 1214605671
```

If you **get** the Sorting, you always get the FieldUID (GUID) as String.

```
Dim s As String
s = CatalogCollection.Sorting' returns "{af4b2e0e-5f6a-11d2-8f20-0000c0e166dc}"
```

In case of an invalid sorting (for example, after moving single records to other positions using MoveRecord), the return string is empty.

The different predefined field id constants are listed in FieldAccessConstants.

See also CatalogCollection

SortDirection As Long

This property returns or sets the sort direction of the current catalog window (collection).

Possible values are:

Ascending	= 0
Descending	= 1

See also SortDirectionConstants, CatalogCollection

Query As String

This property returns or sets a collection's (catalog windows) query in the (Record) Findbar. A query consists of the strings that are available in the (Record) Findbar pulldown-menus or free definable strings for the operands. The single parts are divided by a “Tab” character, the query’s conditions are divided by a “Return” character.

VB sample code:

```
CatalogCollection.Query = "Record Name" & vbTab & "starts with" & vbTab & "0" & vbCr _
 & "or" & vbTab & "Record Name" & vbTab & "contains" & vbTab & "8"
MsgBox CatalogCollection.Query
```

See also CatalogCollection

QueryCategories As String (Enterprise Edition only)

This property returns or sets a collection's (catalog windows) query in the (Category) Findbar. A query consists of the strings that are available in the (Category) Findbar pulldown-menus or free definable strings for the operands. The single parts are divided by a “Tab” character, the query’s conditions are divided by a “Return” character.

VB sample code:

```
CatalogCollection.QueryCategories = "Category Name" & vbTab & "starts with" & vbTab & "mino" & vbCr _
  & "or" & vbTab & "Category Name" & vbTab & "contains" & vbTab & "boxes_d"
MsgBox CatalogCollection.QueryCategories
```

See also CatalogCollection

WindowHandle As Long

This read-only property returns the window handle of the MDIClient Child Window of the collection. You can use this handle to perform any operation on the collection window which is accessible by the handle.

VB sample code:

```
Private Declare Function ShowWindow Lib "user32" (ByVal hwnd As Long, ByVal nCmdShow As Long) As Long
Const SW_MINIMIZE = 6
Dim h As Long

h = coll.WindowHandle
ShowWindow h, SW_MINIMIZE
```

Note: To get access to the category and record pane window, you have to get child windows of the returned window.

See also CatalogCollection

New ([Query As String]) As CatalogCollection

This method creates a new collection of the catalog using the given query and returns the CatalogCollection object.

This method opens a new collection view of the catalog.

If the Query string is omitted, the new catalog view will show all records that are available in the current catalog, unless a search is performed.

If the Query string is empty (""), the new collection won't contain any records ("New Empty Collection")

The syntax of the Query string is described in the Find method.

```
Query = "Record Name" & vbTab & "starts with" & vbTab & "A"

' Creates a new window showing all records:
Set R = CatalogCollection.New

' Creates a new window showing all records whose record name starts with "A"
Set R = CatalogCollection.New(Query)

' Creates a new window showing no records
Set R = CatalogCollection.New("")
```

This query string is language-dependent. Please use the FindToken property to assemble a language-independent query string.

See also CatalogCollection

Find (Query As String, FindType As Long) As Long

This method launches a (record) search. A query consists of the strings that are available in the Findbar pulldown-menus or free definable strings for the operands. The single parts are divided by a “Tab” character, the query’s conditions are divided by a “Return” character.

The method returns the number of found records.

If the given “Query” string is empty, nothing will be found.

The FindType argument specifies the kind of the search:

FindNew	= 0
FindAgain	= 1
FindExtend	= 2

FindNew
This search is performed in all records.

FindAgain
This search will use the previous search result (current view) and searches only in this set of records.

FindExtend
This search is performed in all records, and the result is added to the current view (last search result).

VB sample code:

```
query = "Record Name" & vbTab & "starts with" & vbTab & "0" & vbCr _
 & "or" & vbTab & "Record Name" & vbTab & "contains" & vbTab & "8"
recordsFound = CatalogCollection.Find(query, FindNew)
```

This query string is language-dependent. Please use the FindToken property to assemble a language-independent query string.

See also FindTypeConstants, see also CatalogCollection

FindCategories (Query As String, FindType As Long) As Long (Enterprise Edition only)

This method launches a (category) search. A query consists of the strings that are available in the Category Findbar pulldown-menus or free definable strings for the operands. The single parts are divided by a “Tab” character, the query’s conditions are divided by a “Return” character.

The method returns the number of found categories.

If the given “Query” string is empty, nothing will be found.

The FindType argument specifies the kind of the search:

FindNew	= 0
FindAgain	= 1
FindExtend	= 2

FindNew
This search is performed in all categories.

FindAgain
This search will use the previous search result (current category view) and searches only in this set of categories.

FindExtend
This search is performed in all categories, and the result is added to the current category view (last search result).

VB sample code:

```
query = "Category Name" & vbTab & "starts with" & vbTab & "mino" & vbCr _
  & "or" & vbTab & "Category Name" & vbTab & "contains" & vbTab & "boxes_d"
categoriesFound = CatalogCollection.FindCategories(query, FindNew)
```

This query string is language-dependent. Please use the FindToken property to assemble a language-independent query string.

See also FindTypeConstants, see also CatalogCollection

FindAll

This method shows ALL records of the catalog in the current collection. It reverses the effect of the Find method.

See also CatalogCollection

FindAllCategories (Enterprise Edition only)

This method shows ALL categories of the catalog in the current collection. It reverses the effect of the FindCategories method.

See also CatalogCollection

Application As Application

This property returns the Application object.

See also CatalogCollection

Parent As Object

This property returns the CatalogCollection’s parent object.

See also CatalogCollection

Name As Object

This (read-only) property returns the name of the CatalogCollection. This is the window title of the catalog window (collection). This title consists of the catalog name and can be followed by a colon followed either by a number (in case of unnamed collection) or by the name of the opened Cumulus collection file (for example MyCollection.cfe).

"Name" is the **default** property.

Examples of the composed Name:

MyCatalog
MyCatalog:1
MyCatalog:2
MyCatalog:MyCollection

Note: The extensions ":1", ":2", ... for unnamed collections can change if unnamed collection windows are closed.

See also CatalogCollection

CatalogAsset (PathName As String)

This method catalogs the given file or folder and shows the new records in the collection window.

If you specify an empty string ("") as PathName, a dialog will be shown which allows you to select an asset storage (if necessary) and a folder / asset for the cataloging process.

See also CatalogCollection

Catalog As Catalog

This property returns the collection's catalog object. All collections (views) of the same catalog refer to the same catalog object.

See also CatalogCollection

ImportCategories (FromPathName As String)

This method imports categories from the given category export file (*.cce) that was previously created using the Categories.Export method.

See also CatalogCollection

ImportRecords (FromPathName As String)

This method imports records from the given record export file (*.cre) that was previously created using the Records.Export method.

See also CatalogCollection

Mail

This method creates a temporary collection file (containing the current state of the collection that could be also created using Save) and mails this file as e-mail attachment of one single e-mail. After that, the collection file will be deleted.

Note: Because the-mail client will be started by the VB application as modal, you should switch off the timeout error message. You can do this using the App object's OleRequestPendingTimeout method. The App object (this is not the Cumulus applications object, but the VB program) is a global object accessed with the App keyword. It determines or specifies information about the application.

VB sample code:

```
' switch off the timeout message
Dim oldTimeout As Long
oldTimeout = app.OleRequestPendingTimeout ' save the current value
App.OleRequestPendingTimeout = 2000000000 ' some milliseconds

Collection.Mail ' mail the collection file.

App.OleRequestPendingTimeout = oldTimeout ' restore the timeout value
```

See also CatalogCollection, Records.MailTheAssets

CategoryListVisible As Boolean

This property gets or sets the category visibility state of the collection window. It shows or hides the category pane (left part of the window).

It is not possible to switch off the category pane and the record pane at the same time.

See also CatalogCollection

RecordListVisible As Boolean

This property gets or sets the category visibility state of the collection window. It shows or hides the record pane (right part of the window).

It is not possible to switch off the category pane and the record pane at the same time.

See also CatalogCollection

SelectedCategories As Categories

This property returns the selected categories of the collection window (selected categories in the category pane of the collection).

See also CatalogCollection

SelectedRecords As Records

This property returns the selected records of the collection window (selected records in the record pane of the collection).

See also CatalogCollection

CategoryTree As Categories

This property returns a top level categories object of the collection as a list, sorted by name.

Note: This categories object contains only the **top-level** categories of the collection.

See also CatalogCollection

View As Long

This property returns or sets the collection’s viewing type.

Possible values are:

Details	= 0
SmallThumbnails	= 1
MediumThumbnails	= 2
LargeThumbnails	= 3
CustomSizeThumbnails	= 4
PaletteView	= 5

The values CustomSizeThumbnails and PaletteView are read-only.

If the Cumulus state is Palette View, the "View" property is on principle read-only.

To set a CustomSizeThumbnails thumbnail value, you first have to switch to a predefined thumbnail view size and select the requested size using ThumbnailImageSize.

```
Collection.View = MediumThumbnails
Collection.ThumbnailViewConfiguration.ThumbnailImageSize = 50
```

To switch to PaletteView, you have to use Application.PaletteMode.

```
App.PaletteMode = true
```

See also ViewConstants, CatalogCollection, ViewConfiguration

Save (FileName As String)

This method saves the current view content in a Cumulus collection file (*.cfe).

This collection file can be restored later by using the CatalogCollections.Add method.

See also CatalogCollection

DetailsViewConfiguration As ViewConfiguration

This property returns the ViewConfiguration object that describes the properties of the catalog collection's **details** view (see also ThumbnailViewConfiguration).

See also CatalogCollection

PaletteViewConfiguration As ViewConfiguration

This property returns the ViewConfiguration object that describes the properties of the catalog collection's **palette** view.

See also CatalogCollection

ThumbnailViewConfiguration As ViewConfiguration

This property returns the ViewConfiguration object that describes the properties of the catalog collection's **thumbnail** view (see also DetailsViewConfiguration).

See also CatalogCollection

Categories As Categories

This property returns the categories object of the collection as a list, sorted by name.

See also CatalogCollection

Records As Records

This property returns the records object of the collection .

See also CatalogCollection

Application As Application

This property returns the Application object.

See also Records

MailTheAssets ([IDList As Variant])

This method allows you to e-mail assets (as e-mail attachment of one single e-mail) of the Records object. The behavior of this method depends on the origin of the Records object.

CatalogCollection.Records:
If the optional IDList argument is omitted, all assets that belong to the Records in the record pane of the CatalogCollection will be mailed.

CatalogCollection.SelectedRecords:
If the optional IDList argument is omitted, all assets that belong to the selected records in the record pane of the CatalogCollection will be mailed.

If an IDList argument is provided, all specified records of record pane of the CatalogCollection will be mailed (independent of the origin of the Records object).

Note: Because the e-mail client will be started by the VB application as modal, you should switch off the timeout error message. You can do this using the App object's OleRequestPendingTimeout method. The App object (this is not the Cumulus applications object, but the VB program) is a global object accessed with the App keyword. It determines or specifies information about the application.

VB sample code:

```
Dim Recs As Records

Dim List(1) As Long
List(0) = 1
List(1) = 4

Set Recs = CatalogCollection.Records

' switch off the timeout message
Dim oldTimeout As Long
oldTimeout = app.OleRequestPendingTimeout ' save the current value
App.OleRequestPendingTimeout = 2000000000 ' some milliseconds

Recs.MailTheAssets List ' Mails two assets with the ID 1 and 4

App.OleRequestPendingTimeout = oldTimeout ' restore the timeout value
```

See also Records, CatalogCollection.Mail

CopyTheAssets (ToPathName As String, [IDList As Variant])

This method allows you to copy assets of the Records object.

The argument ToPathName specifies a target Windows directory for the copy process. If you specify an empty string ("") as ToPathName, a dialog will be shown which allows you to select an asset storage (if necessary) and a target location for the copy process.

The behavior of this method depends on the origin of the Records object.

CatalogCollection.Records:
If the optional IDList argument is omitted, all assets that belong to the Records in the record pane of the CatalogCollection will be copied.

CatalogCollection.SelectedRecords:
If the optional IDList argument is omitted, all assets that belong to the selected records in the record pane of the CatalogCollection will be copied.

If an IDList argument is provided, all specified records in the record pane of the CatalogCollection will be copied (regardless of the origin of the Records object).

VB sample code:

```
Dim Recs As Records

Dim List(1) As Long
List(0) = 1
List(1) = 4

Set Recs = CatalogCollection.Records
Recs.CopyTheAssets "C:\MyDir", List ' Copies two assets with the ID 1 and 4
```

See also Records

MoveTheAssets (ToPathName As String, [IDList As Variant])

This method allows to move assets of the Records object.

The argument ToPathName specifies a target Windows directory for the move process. If you specify an empty string ("") as ToPathName, a dialog will be shown which allows you to select an asset storage (if necessary) and a target location for the move process.

During the move process the AssetReferences are updated.

The behavior of this method depends on the origin of the Records object.

CatalogCollection.Records:
If the optional IDList argument is omitted, all assets that belong to the Records in the record pane of the CatalogCollection will be moved.

CatalogCollection.SelectedRecords:
If the optional IDList argument is omitted, all assets that belong to the selected records in the record pane of the CatalogCollection will be moved.

If an IDList argument is provided, all specified records of record pane of the CatalogCollection will be moved (regardless of the origin of the Records object).

VB sample code:

```
Dim Recs As Records

Dim List(1) As Long
List(0) = 1
List(1) = 4

Set Recs = CatalogCollection.Records
Recs.MoveTheAssets "C:\MyDir", List ' Moves two assets with the ID 1 and 4
```

See also Records

ExportToHTML(PathName As String, TemplateNameOrLinkType As Variant, [IDList As Variant])

This method allows you to export the entries of the Records object to an HTML file. The argument PathName specifies a target html-file for the export process. If the file already exists, it will be overwritten.

The TemplateNameOrLinkType argument can be

either a string containing a template name specifying an existing html template that is used for the export

or a long value containing one of the PreviewLinkTypeConstants (in this case the current view settings (details- or thumbnail-view) of the catalog collection are used for the export). Possible values for the link type are:

HtmlNoLink	= 0
HtmlPreviewLink	= 1
HtmlFileLink	= 2

Additionally, you can **add** the following flag constant if you want to activate the "Include Audio Field Link" switch
HtmlFlagIncludeAudioFieldLink   = 256

The behavior of this method depends on the origin of the Records object.

CatalogCollection.Records:
If the optional IDList argument is omitted, all assets that belong to the Records in the record pane of the CatalogCollection will be exported.

CatalogCollection.SelectedRecords:
If the optional IDList argument is omitted, all assets that belong to the selected records in the record pane of the CatalogCollection will be exported.

If an IDList argument is provided, all specified records of record pane of the CatalogCollection will be exported (regardless of the origin of the Records object).

VB sample code:

```
Dim Recs As Records

Dim List(1) As Long
List(0) = 1
List(1) = 4

Set Recs = CatalogCollection.Records

' Export with current view settings with no record link but audio field link
Recs.ExportToHTML "c:\test.html", HtmlNoLink + HtmlFlagIncludeAudioFieldLink

' Export with current view settings and with a given ID list. Exports two records with the ID 1 and 4
Recs.ExportToHTML "c:\test.html", HtmlPreviewLink, List

' Export using the template "T1"
Recs.ExportToHTML "c:\t1.html", "T1"
```

See also Records

Count As Long

The behavior of this property depends on the origin of the Records object.

CatalogCollection.Records:
This property returns the number of records in the record pane of the CatalogCollection.

CatalogCollection.SelectedRecords:
This property returns the number of selected records in the record pane of the CatalogCollection.

See also Records

Parent As Object

This property returns the records' parent object.

See also Records

New As Record

The behavior of this property depends on the origin of the Records object.

CatalogCollection.Records:
This method creates a new empty record (for testing purposes mainly) and returns the newly created record object

CatalogCollection.SelectedRecords:
This method is not available if the Records object is taken from CatalogCollection.SelectedRecords.

See also Records

Add (Record As Record) As Record

The behavior of this property depends on the origin of the Records object.

CatalogCollection.Records:
This method copies a record to the CatalogCollection (it is also added to catalog) and returns the newly created record object. The record may also be taken from another catalog.

CatalogCollection.SelectedRecords:
This method adds the record object to the CatalogCollection's selection in the record pane (the record is selected). The record to be added must be a valid record from the catalog. The return value is always “Nothing” and cannot be used.

See also Records

Item (Index As Long / Name As String) As Record

This property returns a record object by index or by name.

This is the **default** property.

The behavior of this property depends on the origin of the Records object.

CatalogCollection.Records:
The range of index / name covers the complete set of records in the CatalogCollection's record pane.

CatalogCollection.SelectedRecords:
The range of index / name covers the complete set of selected records in the CatalogCollection's record pane.

See also Records

ItemByID (ID As Long) As Record

This property returns a record object by record ID.

The behavior of this property depends on the origin of the Records object.

CatalogCollection.Records:
The range of valid IDs covers the complete set of records in the CatalogCollection's record pane.

CatalogCollection.SelectedRecords:
The range of valid IDs covers the complete set of selected records in the CatalogCollection's record pane.

See also Records

Remove (Index As Long / Name As String)

The behavior of this property depends on the origin of the Records object.

CatalogCollection.Records:
This method deletes a record from the catalog by index or by name. If a name is given which specifies more than one record, only the first one is deleted. The range of index / name covers the complete set of records in the CatalogCollection's record pane.

CatalogCollection.SelectedRecords:
This method removes a record from the CatalogCollection’s record selection by index or by name (deselects the record). If a name is given which specifies more than one record, only the first one is removed. The range of index / name covers the complete set of selected records in the CatalogCollection's record pane.

See also Records

RemoveByID (ID As Long)

The behavior of this property depends on the origin of the Records object.

CatalogCollection.Records:
This method deletes a record from the catalog by Record ID. If a name is given which specifies more than one record, only the first one is removed. The range of valid IDs covers the complete set of records in the CatalogCollection's record pane.

CatalogCollection.SelectedRecords:
This method removes a record from the CatalogCollection’s record selection by Record ID (deselects the record). If a name is given which specifies more than one record, only the first one is removed. The range of valid IDs covers the complete set of selected records in the CatalogCollection's record pane.

See also Records

Export (ToPathName As String [, UTF8 As Boolean])

Export exports a Cumulus catalog’s records. If the export file does not exist it is created, else overwritten.

If you specify the optional argument UTF8 = True, the export file will be encoded in the UTF8 format (local character set independent format).

The behavior of this property depends on the origin of the Records object.

CatalogCollection.Records:
Export exports the information of all records contained in the records object (all records in the records list of the CatalogCollection).

CatalogCollection.SelectedRecords:
Export exports the information of all selected records in the record pane of the CatalogCollection.

See also Records

Count As Long

This property returns the number of CatalogCollection objects. This is the number of all open collections.

See also CatalogCollections

Parent As Object

This property returns the CatalogCollections' parent object.

See also CatalogCollections

Item (Index As Long / Name As String) As CatalogCollection

This property returns a CatalogCollection object by index or name (window title).

This is the **default** property.

The returned object represents one collection view.

See also CatalogCollections

ItemByID (ID As Long) As CatalogCollection

This property returns a CatalogCollection object by ID.

The returned object represents one collection view.

See also CatalogCollections

Remove (Index As Long / Name As String)

This method removes a CatalogCollection object from the catalog collections by index or name (window title).

This method closes the specified collection of a catalog. If the last view of a catalog is closed, the whole catalog will be closed.

See also CatalogCollections

RemoveByID (ID As Long)

This method removes a CatalogCollection object from the catalog collections by Collection ID.

This method closes the specified collection of a catalog. If the last view of a catalog is closed, the whole catalog will be closed.

See also CatalogCollections

Application As Application

This property returns the Application object.

See also CatalogCollections

Add (FileName As String) As CatalogCollection

This method creates a new CatalogCollection object and adds it to the CatalogCollections object.

The specified File (FileName) must be a Cumulus collection file (*.cfe) that was previously saved using CatalogCollection.Save.

This method opens a new catalog window (collection view). The content (visible records, etc.) is taken from the information in the Cumulus collection file.

See also CatalogCollections

AllowSharing As Boolean (Administrator only)

This property returns or sets a remote catalog’s sharing state.

See also Catalog, ServerCatalog.IsShared

Application As Application

This property returns the Application object.

See also Catalog

Unlock (Password As String) (Administrator only)

If a catalog is locked (this can be checked using the IsLocked property), you can unlock it with the Unlock method. You have to specify the correct Password to unlock the catalog.

See also Catalog

IsLocked As Boolean

This property returns if the catalog (database) is write-protected because it has a write access password and was opened without using the correct password or explicitly as "Read-Only". You can use the Unlock method to unlock the catalog with the password.

See also Catalog

IsReadOnly As Boolean

This property returns if the catalog (database) is physically read-only (for example if the Catalog.ccf file is read-only).

See also Catalog

WriteAccessPassword (Password As String) (Administrator only)

If the catalog is NOT locked with a password (no password set or unlocked with the Unlock method), you can use this property to get the current password or to set a new one.

See also Catalog

CanDeleteRecords As Boolean

This property returns if the current user is allowed to delete records.

See also Catalog

CentralAssetLocationXML As String (Administrator only)

This property returns or sets the central asset location reference as XML. This property can return any asset reference whereas the property Central Asset Location only returns a pathname of a Windows folder.

See also Catalog

FindAssetXML (AssetReferenceXML As String) As Variant

This method returns a list of record IDs (array of long) for all records that reference the given asset.

See also Catalog

AssetInformationConfiguration As InformationConfiguration

This property returns the InformationConfiguration object that describes the properties of the catalog's asset information window (see also CategoryInformationConfiguration).

See also Catalog

CategoryInformationConfiguration As InformationConfiguration

This property returns the InformationConfiguration object that describes the properties of the catalog's category information window (see also AssetInformationConfiguration).

See also Catalog

CanModifyCategories As Boolean

This property returns if the current user is allowed to modify the category list (adding, changing or deleting categories).

See also Catalog

CanModifyRecords As Boolean

This property returns if the current user is allowed to modify records.

See also Catalog

CanTransferAssets As Boolean

This property returns if the current user is allowed to transfer assets.

See also Catalog

CatalogServer As String

This property returns the catalog server's name if the catalog is accessed as a client. For directly opened catalogs, the returned string is empty.

See also Catalog

CentralAssetLocation As String (Administrator only)

This property returns or sets the path of the catalog’s central asset location folder.

If you specify an empty string ("") as the setting a new central asset location folder, a dialog will be shown which allows you to select an asset storage (if necessary) and a folder.

See also Catalog

CentralAssetLocationUsage As Long (Administrator only)

This property returns or sets the current CentralAssetLocation usage value.

Possible values are:

Always	= 0
AsCatalogingOption	= 1
Never	= 2

See also CentralAssetLocationUsageConstants, Catalog

FullName As String (Administrator only)

This property returns the full pathname of the catalog.

See also Catalog

Groups As Groups (Enterprise Edition and Administrator only)

This property returns the catalog's groups collection.

See also Catalog

IsRemoteCatalog As Boolean

This property returns True if the catalog is opened as a client.

See also Catalog

IsAdmin As Boolean

This property returns True if the catalog is opened locally (Single User) or as a client with administrator access.

See also Catalog

KeepCategoryNamesUnique As Boolean (write access: Administrator only)

This property returns or sets if duplicate category names are allowed for this catalog.

See also Catalog

KeepDeleted As Boolean (Administrator only)

This property returns or sets if deleted records should be kept for a later recovery or if they should be deleted permanentely.

See also Catalog

Name As String (write access: Administrator only)

This property returns or sets the catalog’s name.

This is the **default** property.

See also Catalog, ServerCatalog.Name

ID As Long

This property returns the catalog’s unique ID. This ID is valid until the catalog is closed. If the same catalog is reopened, it will have another ID.

See also Catalog

ConvertTo (PathName As String [, DatabaseSelectorUID As String]) (Administrator only)

This method creates a catalog with the given PathName. This catalog is a copy of the current **remote** catalog. You can use this method to create a copy of the catalog using **another** database engine.

The argument DatabaseSelectorUID (a GUID string value) allows you to select a specific database engine. If you omit this argument or set it to "", the Cumulus Engine is used as default. Some possible values are:

UID_SELECTOR_CUE (Cumulus Engine)
UID_SELECTOR_ODBC (ODBC Connector)

It is allowed to specify the name of the database selector instead of the UID (for example "Cumulus Database Engine" or "ODBC Database Connector").

See also Catalog, FieldAccessConstants

WriteMetadataToAssets As Long (Administrator only)

This property returns or sets the state of the "Writing Metadata To Assets" value.

Possible values are:

WriteMetadataNever	= 0
WriteMetadataSilent	= 1
WriteMetadataWithUI	= 2

See also Catalog, WriteMetadataConstants

SendMessage (ServerModule As String, Message As String [, Parameters As String]) As String (Administrator only)

This method sends a message to a server plug-in module. See the documentation of the corresponding server module for a description of the messages, parameters and reply strings.

ServerModule:	the name or unique ID of the server module
Message:	the message string
Parameters:	optional parameter string

This method returns a reply string.

See also Catalog

CatalogReference As String (Administrator only)

This read-only property returns the reference data for this catalog. The returned string contains the binary data as a hex-character representation.

This value can be used to open the catalog via Application.OpenCatalogByReference.

VB sample code:

```
MsgBox catalog.CatalogReference
```

shows "6F63657205000000440000006769724F6F6365720200000010000000044496C4364695567D07AD375B048D211AB9F008048E82F961800000053636F4C7268435563003A005C006D00610072006B006500740069006E00670010000000044496C4364695567D07AD375B048D211AB9F008048E82F961A00000069"

This represents 196 bytes of binary data (even if the string has the double length).

See also Catalog

NativeThumbnailSize As Long (write access: Administrator only)

This property returns or sets the native thumbnails size in pixels of the catalog, which is used for cataloging assets. The allowed range is from 16 to 1024.

If you want to set a current value for large, medium or small thumbnails you can request them using Preferences.ThumbnailSize.

See also Catalog

Parent As Object

This property returns the catalog's parent object.

See also Catalog

RecordLayout As Layout

This property returns the Record Layout of a catalog.

See also Catalog

CategoryLayout As Layout

This property returns the Category Layout of a catalog.

See also Catalog

UserName As String

This property returns the name of the user who has connected (as a registered user with a login name) to a server and opened this catalog shared by the server. If the user is "guest," an empty string will be returned.

For directly opened catalogs, the UserName is the user login name (on the local computer).

See also Catalog

RecordsDeleted As Long (Administrator only)

This property returns the number of deleted records. These records can be reclaimed by calling RecoverDeletedRecords.

See also Catalog

RestrictedPermissions As Boolean (Administrator only)

This property returns or sets if the catalog has restricted user permissions.

See also Catalog

UserPermissions As UserPermissions (Administrator only)

This property returns the catalog's user permissions collection.

See also Catalog

Backup (PathName As String) (Administrator only)

This method stores a backup copy of the catalog in the given file.

See also Catalog

NewEmptyCopy (PathName As String [, DatabaseSelectorUID As String]) As CatalogCollection (Administrator only)

This method creates a new empty catalog with the given PathName and returns the CatalogCollection object.

The new catalog is empty (no records and no categories) and has the same layout (fields and properties) as the original catalog.

The argument DatabaseSelectorUID (a GUID string value) allows you to select a specific database engine. If you omit this argument or set it to "", the Cumulus Engine is used as default. Some possible values are:

UID_SELECTOR_CUE (Cumulus Engine)
UID_SELECTOR_ODBC (ODBC Connector)

It is allowed to specify the name of the database selector instead of the UID (for example "Cumulus Database Engine" or "ODBC Database Connector").

See also Catalog, FieldAccessConstants

SetAsDefault (Administrator only)

This method stores the current catalog layout (fields and properties) as default values for the creation of new catalogs. If you create a new catalog, it will have this layout. See also Application.RestoreFactoryDefaults.

See also Catalog

Compress (Administrator only)

This method compresses the catalog. If the catalog is compressed, previously deleted records cannot be restored using RecoverDeletedRecords.

See also Catalog

Rebuild (Administrator only)

This method repairs the catalog and rebuilds the indices.

See also Catalog

RecoverDeletedRecords (Administrator only)

This method recovers all deleted records from the catalog.

See also Catalog

PublishToWWW As Boolean (Administrator only)

This property returns or sets if the catalog is published to the World Wide Web.

See also Catalog, ServerCatalog.IsPublishedToInternet

AddDuplicateAssets As Boolean

This property returns or sets if duplicates of files should be cataloged or not.

See also CatalogingOptions

Application As Application

This property returns the Application object.

See also CatalogingOptions

CatalogingAction As Long

This property returns or sets the action, that is used for the cataloging process.

Possible values are:

AddOnly	= 0
AddOrUpdate	= 1
UpdateOnly	= 2

See also CatalogingActionConstants, CatalogingOptions

ColorThumbnailQuality As Long

This property returns or sets the thumbnail quality for creating color thumbnails.

Possible values are:

NormalQuality	= 0
HighQuality	= 1

See also ThumbnailQualityConstants, CatalogingOptions

CopyAssetToCentralLocation As Boolean

This property returns or sets if the file should be copied to the selected central location folder before cataloging or if the original file should be referenced.

See also CatalogingOptions

ShowProgressBar As Boolean

This property returns or sets if the files of a folder or volume should be counted before cataloging to show a more detailed progress bar.

See also CatalogingOptions

ExcludeDamagedAssets As Boolean

This property returns or sets if damaged assets (that might possibly cause Cumulus to crash) should be added to the exclude asset list automatically while cataloging.

See also CatalogingOptions

ExcludedAssets As List of String

This property returns or sets a list of excluded assets which will not be cataloged.

The following sample code demonstrates the usage:

```
' Set the excluded assets
Dim ex(1) As String
ex(0) = "c:\autoexec.bat"
ex(1) = "c:\config.sys"
App.CatalogingOptions.ExcludedAssets = ex

' Get the excluded assets
Dim v As Variant
Dim size As Integer
Dim i As Integer

v = App.CatalogingOptions.ExcludedAssets
size = UBound(v) - LBound(v) + 1
MsgBox size
For i = 0 To size - 1
    MsgBox v(i)
Next
```

You can clear all complete list of excluded asset categories using the following statements

```
Dim ex(0) As String
App.CatalogingOptions.ExcludedAssets = ex ' Clears all excluded asset categories
```

See also CatalogingOptions

AssetFormatTable As AssetFormatTable

This property returns the AssetFormatTable.

See also CatalogingOptions

AssetStorages As AssetStorages

This read-only property returns the AssetStorages object.

See also CatalogingOptions

GrayscaleThumbnailQuality As Long

This property returns or sets the thumbnail quality for creating grayscale thumbnails.

Possible values are:

NormalQuality	= 0
HighQuality	= 1

See also ThumbnailQualityConstants, CatalogingOptions

LineartThumbnailQuality As Long

This property returns or sets the thumbnail quality for creating lineart thumbnails.

Possible values are:

NormalQuality	= 0
HighQuality	= 1

See also ThumbnailQualityConstants, CatalogingOptions

OpenPropertyWindow As Boolean

This property returns or sets if after cataloging a new asset the property window should be opened automatically to allow immediate editing.

See also CatalogingOptions

OptimizeThumbnailsAutomatically As Boolean

This property returns or sets if the thumbnails should be automatically optimized in brightness and contrast during cataloging.

See also CatalogingOptions

Parent As Object

This property returns the cataloging option’s parent object.

See also CatalogingOptions

ResolveShortcuts As Boolean

This property returns or sets if shortcuts should be resolved or ignored during cataloging.

See also CatalogingOptions

CatalogAllFormats As Boolean

This property returns or sets if ALL (!) formats that are available should be used for cataloging (CatalogAllFormats = True) or only those that are activated in the AssetFormatTable (CatalogAllFormats = False).

See also CatalogingOptions

ThumbnailSource As Long

This property returns or sets the wanted source for the created thumbnails.

Possible values are:

FromEntireAsset	= 0
FromStoredThumbnail	= 1
FromStoredThumbnailIfLargeEnough	= 2

See also ThumbnailSourceConstants, CatalogingOptions

New As Category

The behavior of this method depends on the origin of the Categories object.

CatalogCollection.Categories:
CatalogCollection.CategoryTree:
This method adds (creates) a new category on the top level of a Cumulus catalog using the default category name and returns the newly created category object.

Category.SubCategories:
This method adds (creates) a new category as subcategory of the (parent)category using the default category name and returns the newly created category object.

CatalogCollection.SelectedCategories:
This method is not available if the Categories object is taken from CatalogCollection.SelectedCategories.

Record.Categories:
This method is not available if the Categories object is taken from Record.Categories.

See also Categories

AddRecordCategory (ID As Long)

The behavior of this method depends on the origin of the Categories object.

CatalogCollection.Categories:
CatalogCollection.CategoryTree:
Category.SubCategories:
CatalogCollection.SelectedCategories:
This method is not available if the Categories object is taken from these sources.

Record.Categories:
This method adds a category to the categories assigned to the record object. The category (described by the ID) to be added must be a valid category from the catalog.
Note: After adding all needed categories to the record object you have to call the Record.Save method to store the changes permanently in the catalog.

See also Categories

Add (Category As Category) As Category

The behavior of this method depends on the origin of the Categories object.

CatalogCollection.Categories:
CatalogCollection.CategoryTree:
This method copies a category including all subcategories and original categories on the top level of a Cumulus catalog and returns the newly created category object. The Category may also be taken from another catalog.

Category.SubCategories:
This method copies a category including all subcategories and original categories as subcategory of the (parent)category and returns the newly created category object. The Category may also be taken from another catalog.

CatalogCollection.SelectedCategories:
This method adds a category to the category selection (in the category pane) of the CatalogCollection (the category is selected). The category to be added must be a valid category from the catalog. The return value is always “Nothing” and cannot be used.

Record.Categories:
This method adds a category to the categories assigned to the record object. The category to be added must be a valid category from the catalog. The return value is always “Nothing” and cannot be used.
Note: After adding all needed categories to the record object you have to call the Record.Save method to store the changes permanently in the catalog.

See also Categories

Application As Application

This property returns the Application object.

See also Categories

CategoryNames (IDList As Variant) As Variant

This read-only property returns for a given Category ID List a list of strings containing the corresponding category names. You can use this function if you need more performance than iterating through the single Categories and request the names using the Item / fields / field / value hierarchy.

VB sample code:

```
Dim i As Integer

Dim Recs As Records
Dim r As Record
Dim rcats As categories

' Take the collection "coll" from anywhere...
Set Recs = coll.Records
Set r = Recs.Item(0)

Set rcats = r.categories

Dim IDs() As Long
ReDim IDs(rcats.count - 1)

' Fill the Category ID List
For i = 0 To rcats.count - 1
    IDs(i) = rcats.Item(i).ID
Next

' Get all names with one single call
Dim Names() As String
Names = rcats.CategoryNames(IDs)

' Display the result
For i = 0 To rcats.count - 1
    MsgBox Names(i)
Next
```

See also Categories

Count As Long

The behavior of this property depends on the origin of the Categories object.

CatalogCollection.Categories:
This property returns the number of categories in the catalog.

Category.SubCategories:
This property returns the number of direct subcategories of the (parent)category.

CatalogCollection.CategoryTree:
This property returns the number of root (top level) categories in the catalog.

CatalogCollection.SelectedCategories:
This property returns the number of selected categories in the category pane of the CatalogCollection.

Record.Categories:
This property returns the number of categories assigned to the record object.

See also Categories

Export (ToPathName As String [, UTF8 As Boolean])

Export exports the categories of the Cumulus catalog. If the export file does not exist, it is created, otherwise it is overwritten.

If you specify the optional argument UTF8 = True, the export file will be encoded in the UTF8 format (local character set independent format).

The behavior of this property depends on the origin of the Categories object.

CatalogCollection.Categories:
All Categories of the catalog are exported. Use ExportVisiblesOnly if you want to export only the categories that are visible in the pane (not collapsed).

Category.SubCategories:
This method is not available if the Categories object is taken from Category.SubCategories.

CatalogCollection.CategoryTree:
This method is not available if the Categories object is taken from CatalogCollection.CategoryTree.

CatalogCollection.SelectedCategories:
All selected categories (AND their subcategories) in the category pane of the CatalogCollection are exported.

Record.Categories:
All categories that are assigned to the record (AND their subcategories) are exported.

See also Categories, ExportVisiblesOnly

ExportVisiblesOnly (ToPathName As String [, UTF8 As Boolean])

Export exports the visible (not collapsed) categories of the category pane. If the export file does not exist, it is created, otherwise it is overwritten.

If you specify the optional argument UTF8 = True, the export file will be encoded in the UTF8 format (local character set independent format).

The behavior of this property depends on the origin of the Categories object.

CatalogCollection.Categories:
All **visible** (not collapsed) Categories of the category pane are exported. Use Export to export **all** categories

This method is not available if the Categories object is taken from
Category.SubCategories:
CatalogCollection.CategoryTree:
CatalogCollection.SelectedCategories:
Record.Categories

Use Export in these cases.

See also Categories, Export

Item (Index As Long / Name As String) As Category

This property returns a category object by index or by name.

This is the **default** property.

The behavior of this property depends on the origin of the Categories object.

CatalogCollection.Categories:
The range of index / name covers the complete set of categories in the catalog.

Category.SubCategories:
The range of index / name covers the complete set of direct subcategories of the (parent)category.

CatalogCollection.CategoryTree:
The range of index / name covers the complete set of all root (top level) categories in the catalog.

CatalogCollection.SelectedCategories:
The range of index / name covers the complete set of all selected categories in the category pane of the CatalogCollection.

Record.Categories:
The range of index / name covers the complete set of all categories assigned to the record.

See also Categories

ItemByID (ID As Long) As Category

This property returns a category object by category ID.

The behavior of this property depends on the origin of the Categories object.

CatalogCollection.Categories:
The range of valid IDs covers the complete set of categories in the catalog.

Category.SubCategories:
The range of valid IDs covers the complete set of direct subcategories of the (parent)category.

CatalogCollection.CategoryTree:
The range of valid IDs covers the complete set of all root (top level) categories in the catalog.

CatalogCollection.SelectedCategories:
The range of valid IDs covers the complete set of all selected categories in the category pane of the CatalogCollection.

Record.Categories:
The range of valid IDs covers the complete set of all categories assigned to the record.

See also Categories

Parent As Object

This property returns the categories’ parent object.

See also Categories

Remove (Index As Long / Name As String)

The behavior of this method depends on the origin of the Categories object.

CatalogCollection.Categories:
This method removes a category from a Cumulus catalog by index or by name. The range of index / name covers the complete set of categories in the catalog.

Category.SubCategories:
This method removes a category from a Cumulus catalog by index or by name. The range of index / name covers the complete set of direct subcategories of the (parent)category.

CatalogCollection.CategoryTree:
This method removes a category from a Cumulus catalog by index or by name. The range of index / name covers the complete set of all root (top level) categories of the catalog.

CatalogCollection.SelectedCategories:
This method removes a category from the CatalogCollection's category selection (deselects the category). The range of index / name covers the complete set of all selected categories in the category pane of the CatalogCollection.

Record.Categories:
This method removes a category from the categories assigned to the record. The range of index / name covers the complete set of all categories assigned to the record.
Note: After removing all wanted categories from the record object you have to call the Record.Save method to store the changes permanently in the catalog.

See also Categories

RemoveByID (ID As Long)

The behavior of this method depends on the origin of the Categories object.

CatalogCollection.Categories:
This method removes a category from a Cumulus catalog by category ID. The range of valid IDs covers the complete set of categories in the catalog.

Category.SubCategories:
This method removes a category from a Cumulus catalog by category ID. The range of valid IDs covers the complete set of direct subcategories of the (parent)category.

CatalogCollection.CategoryTree:
This method removes a category from a Cumulus catalog by category ID. The range of valid IDs covers the complete set of all root (top level) categories of the catalog.

CatalogCollection.SelectedCategories:
This method removes a category from the CatalogCollection's category selection (deselects the category). The range of valid IDs covers the complete set of all selected categories in the category pane of the CatalogCollection.

Record.Categories:
This method removes a category from the categories assigned to the record. The range of valid IDs covers the complete set of all categories assigned to the record.
Note: After removing all wanted categories from the record object you have to call the Record.Save method to store the changes permanently in the catalog.

See also Categories

Application As Application

This property returns the Application object.

See also Category

IsReadOnly As Boolean

This property returns if this category is read-only. If a category is read-only, you cannot change its fields and content.

A category is read-only if
a) the user does not have the permission to modify the category list
b) the category is write protected because the user has an active category filter (see UserPermission.CategoryFilter)
c) the catalog is password write protected
d) the catalog (database) is physically read-only (for example if the Catalog.ccf file is read-only).

See also Category

CategoryLayout As Layout

This property returns the category layout of the category. This is also the category layout of the catalog.

See also Category

Collapsed As Boolean

This property returns or sets the collapsed state of the category in the user interface.

See also Category

Fields As Fields

This property returns the Fields object of the category. Using the fields object, you can get or set the single Fields of a category.

This is the **default** property.

See also Category

ID As Long

This property returns the category's internal ID.

See also Category

CollectionIndex As Long

This property returns the Index of the category regarding the CatalogCollection.Categories object.

Note: This is NOT necessarily the index of the Category's parent object like CatalogCollection.CategoryTree, Record.Categories, Category.SubCategories, CatalogCollection.SelectedCategories)!

See also Category

Level As Long

This property returns the category’s level in the category tree.

See also Category

Original As Category

This property returns or sets a related category’s original category.

See also Category

Parent As Object

This property returns the category’s parent object.

See also Category

ParentCategory As Category

This property returns or sets the category's parent category. The value “Nothing” means top level category.

You can check the value "Nothing" like this:

```
Dim cat As Category
Dim c1 As Category
...
Set cat = c1.ParentCategory
If cat Is Nothing Then
            MsgBox "Root Category"
        Else
            MsgBox cat.Name
        End If
```

See also Category

ShowInformationWindow

This method shows the category's information window.

See also Category

SubCategories As Categories

This property returns the collection of direct (next recursion level) Categories of the category.

See also Category

Export (ToPathName As String [, UTF8 As Boolean])

Export exports a single category. If the export file does not exist, it is created, otherwise it is overwritten.

If you specify the optional argument UTF8 = True, the export file will be encoded in the UTF8 format (local character set independent format).

See also Category

Application As Application

This property returns the Application object.

See also AssetFormatEntry

Extension As String

This property returns the extension of the entry's supported asset format.

See also AssetFormatEntry

FileFormat As String

This property returns the file format string.

This is the **default** property.

See also AssetFormatEntry

IsActive As Boolean

This property returns or sets if the entry is activated or inactive.

See also AssetFormatEntry

Name As String

This property returns the entry's control name, for example “Cumulus 4 Compatibility Filter”.

See also AssetFormatEntry

Parent As Object

This property returns the entry's parent object.

See also AssetFormatEntry

Application As Application

This property returns the Application object.

See also AssetFormatTable

Count As Long

This property returns the number of installed asset format entries.

See also AssetFormatTable

MoveAssetFormatEntry (FromIndex As Long, ToIndex As Long)

This method moves an asset format entry from the FromIndex position to the given ToIndex position. In this sense, the filter sequence is changed.

See also AssetFormatTable

Item (Index As Long / Name As String) As AssetFormatEntry

This property returns an AssetFormatEntry by index or by name. If the name consists only of the extension, the first matching asset format entry will be returned; if the name consists of extension plus divider plus the filter name, the specified entry will be returned. The divider can be one of the following characters ‘ ’,‘:’ or ‘-’.

This is the **default** property.

Example:

```
Dim filter As AssetFormatEntry
Dim Table As AssetFormatTable

Set Table = theApp.CatalogingOptions.AssetFormatTable

Set filter = Table.Item("eps:Cumulus 4 Compatibility Filter")
Set filter = Table.Item("lwf")
Set filter = Table.Item(0)
```

See also AssetFormatTable

Parent As Object

This property returns the asset format table's parent object.

See also AssetFormatTable

Application As Application

This property returns the Application object.

See also AssetReference

Parent As Object

This property returns the asset reference’s parent object.

See also AssetReference

Path As String

This property returns the asset reference’s path name as a string.

This is the **default** property.

See also AssetReference

BinaryData As Variant

This property allows you to get / set the AssetReference as binary data. You should use this property only if you have information about the structure of the data. Use extreme caution when modifying this data directly.

After changing the data, you have to call the Record.Save method to save the record’s changes permanently to the catalog's database.

VB sample code:

```
Dim refs As AssetReferences
Dim ref As AssetReference
Dim size As Integer
Dim b() As Byte

Set refs = Record.AssetReferences ' Record taken from anywhere
Set ref = refs.Item(0)

b = ref.BinaryData ' get the data
size = UBound(b) - LBound(b) + 1

' modify the data here
...
' write back and save to the catalog
ref.BinaryData = b
Record.Save
```

See also AssetReference

AssetStorageUID As String

This read-only property returns the ID of the asset storage (a GUID string value like UID_AS_WIN_FILE, UID_AS_MAC_FILE, UID_AS_URL) of this asset reference entry.

The AssetStorageUID can be used to identify the asset storage for this asset reference. The asset storage is responsible for the asset access.

See also AssetReference, FieldAccessConstants

Name As String

This property returns the asset reference’s type as string.

It can be “MacOS” or “Windows” in the current version.

See also AssetReference

Level As Long

This read-only property returns the level of the asset reference. The level starts at 0 for the top level and is counting upward. Assets that are located in other assets have references with levels higher than 0.

See also AssetReference

Application As Application

This property returns the Application object.

See also AssetReferences

Parent As Object

This property returns the asset references’ parent object.

See also AssetReferences

Count As Long

This property returns the count of different asset references for different platforms.

See also AssetReferences

Item (Index As Long / Name As String) As AssetReference

This property returns an asset reference by index or by name.

The name can be “MacOS” or “Windows” in the current version.

This is the **default** property.

See also AssetReferences

Update (PathName As String)

This method updates the asset references with the given path.

See also AssetReferences

XML As String

This property returns the asset references of the record as XML. Setting this property changes all asset reference information of this record.

See also AssetReferences

Application As Application

This property returns the Application object.

See also Preferences

ManualPreviewSize As Long

This property returns or sets the preview size in pixels if “ManualPreviewSize” is chosen as default preview size.

See also Preferences

Parent As Object

This property returns the preferences' parent object.

See also Preferences

PreviewSize As Long

This property returns or sets the default size for opening preview windows.

Possible values are:

OriginalPreviewSize	= 0
MaximumPreviewSize	= 1
ManualPreviewSize	= 2

See also PreviewSizeConstants, Preferences

ResolveRelatedCategories As Boolean

This property returns or sets if related categories should be resolved for searching.

See also Preferences

ThumbnailSize (ViewType As Long) As Long

This property returns or sets the pixel size for one of following three basic thumbnail sizes (specified by ViewType).

Possible values are:

SmallThumbnails	= 1
MediumThumbnails	= 2
LargeThumbnails	= 3

Note: The values "Details", "CustomSizeThumbnails" and "PaletteView" are not allowed for this property.

The allowed range for the ThumbnailSize is from 16 to 1024.

See also Preferences, ViewConstants

SearchForCategoriesAbove As Boolean

This property returns or sets if the category search should also find categories above.

See also Preferences

SearchForCategoriesBelow As Boolean

This property returns or sets if the category search should also find categories below.

See also Preferences

SearchAllSelectedCategories As Boolean

This property returns or sets if the category search should find records contained in all categories or at least one.

See also Preferences

ShowWarnings As Boolean

This property returns or sets if warnings should be shown.

See also Preferences

ThumbnailSimilarityThreshold As Variant

This property returns or sets how the similarity search will find records. The default values are {10, 30}. Use smaller values to do a narrower search.

VB sample code:

```
Dim v As Variant
Dim size As Integer

' Read the Threshold
v = Preferences.ThumbnailSimilarityThreshold
size = UBound(v) - LBound(v) + 1 ' get the size
MsgBox size ' show the size of array
MsgBox v(0)
MsgBox v(1)

' Write the Threshold
v(0) = 11
v(1) = 31
Preferences.ThumbnailSimilarityThreshold = v
```

See also Preferences

Units As Long

This property returns or sets the display unit.

Possible values are:

uInch	= 0
uCentimeter	= 1
uMillimeter	= 2
uPoint	= 3
uPixel	= 4
uPica	= 5

See also UnitConstants, Preferences

UseGrid As Boolean

This property returns or sets if a grid should be used for resizing catalog windows.

See also Preferences

WindowToOpenOnDoubleClick As Long

This property returns or sets what sort of window should be

opened on double clicking a record.

Possible values are:

PropertyWindow	= 0
PreviewWindow	= 1

See also OnDoubleClickConstants, Preferences

Application As Application

This property returns the Application object.

See also Record

CheckIn ([AsUser As String, Comment As String])

This method uses a qualified Asset Storage (if installed and activated) to "check in" a previously checked-out asset (associated with the record).The asset is removed from the check-out folder and is stored back in the Asset Storage's check-in location.

The optional arguments AsUser and Comment can be used to set the user and comment information for the check in history list. If these arguments are omitted, a dialog appears and enables the user to enter this information.

See also Record, CheckOut, UndoCheckOut

CheckOut (CheckOutFolder As String) As String

This method uses a qualified Asset Storage (if installed and activated) to "check out" an asset (associated with the record). The CheckOutFolder argument specifies the target directory to which the asset is copied.

The return value contains the location (Windows Path) of the checked-out file.

VB sample code:

```
MsgBox Record.CheckOut("g:\Check Out Folder") ' returns for example "g:\Check Out Folder\FileXYZ.bmp"
```

See also Record, CheckIn, UndoCheckOut

UndoCheckOut

This method reverts a previously done "check in" operation. The asset (associated with the record) is removed from the check-out folder. If the asset was modified while it was checked out, these changes are lost.

See also Record, CheckIn, CheckOut

MailTheAsset

This method allows you to mail the record's asset (as an e-mail attachment to one single e-mail).

See also Record

IsReadOnly As Boolean

This property returns if this record is read-only. If a record is read-only, you cannot change its fields and content.

A record is read-only if
a) the user does not have the permission to modify records
b) the catalog is password write protected
c) the catalog (database) is physically read-only (for example if the Catalog.ccf file is read-only).

See also Record

ShowLocation

This method opens the Windows Explorer (or another adequate program) to show the record's asset location.

See also Record

EditTheAsset (ApplicationName As String)

This method opens the record's asset with the specified application for editing. If you specify an empty string (""), a file open dialog will be shown.

The ApplicationName can be requested from Record.AssociatedProgram or Application.UsedApplicationList.

You have **four** possibilities to specify the application name:

VB sample code:

```
Dim Apps() As String
Dim Record As Record
Apps = Application.UsedApplicationList
Set Record = Recs(0) ' Take the first Record

' You can specify a windows path:
Record.EditTheAsset "d:\xyz.exe"

' Or an entry of the application list:
Record.EditTheAsset Apps(2)

' Or the associated program:
Record.EditTheAsset Record.AssociatedProgram

' Or an empty string that opens a file open dialog
Record.EditTheAsset ""
```

See also Record, PrintTheAsset

PrintTheAsset (ApplicationName As String)

This method prints the record's asset with the specified application.

For a detailed description of how to specify the application name see EditTheAsset.

See also Record

AssociatedProgram As String

This read-only property returns the record asset's associated program name (only the name, not the path of the executable). This could be, for example, "mspaint" for Windows Bitmap (*.bmp) assets.

This value can be used as the ApplicationName argument for Record.PrintTheAsset or Record.EditTheAsset.

See also Record

Categories As Categories

This property returns the collection of the record’s assigned categories.

See also Record

AssetReferences As AssetReferences

This property returns the asset file's asset references for the different platforms.

See also Record

ID As Long

This property returns the record’s internal ID.

See also Record

CollectionIndex As Long

This property returns the Index of the record regarding the CatalogCollection.Records object.

Note: This is NOT necessarily the index of the Record's parent object like CatalogCollection.SelectedRecords)!

See also Record

Asset As String

This property returns or sets the record’s file display path (unresolved).

If you set the path, you have to call the Record.Save method to store the change permanently in the catalog file:

VB sample code:

```
Dim rec as Cumulus5.Record
…
MsgBox rec.Asset
rec.Asset = "C:\Pictures\Test.jpg"
rec.Asset.Save
```

See also Record

Fields As Fields

This property returns the Fields object of the record. Using the fields object, you can get or set the single Fields of a record.

This is the **default** property.

VB sample code:

```
MsgBox Record.Fields.Item("Record Name")
'shorter, using the default property:
MsgBox Record("Record Name")

MsgBox Record.Fields.ItemByID(UID_REC_RECORD_NAME)
'shorter, using the default property:
MsgBox Record().ItemByID(UID_REC_RECORD_NAME)
```

See also Record

Parent As Object

This property returns the record’s parent object.

See also Record

UpdateAssetReference

This method checks and updates the record’s asset file’s asset reference if the resolver modules can resolve the reference without showing a dialog.

See also Record

CopyTheAsset (ToPathName As String)

This method copies the record’s asset file to the given folder path.

You can specify only a Windows destination **folder name**, but not a new file name. If ToPathName is an empty string (""), a dialog will be prompted and you can select an asset storage (if necessary) and a target location.

See also Record

Delete

This method deletes the record from the catalog.

See also Record

Export (ToPathName As String [, UTF8 As Boolean])

Export exports the record’s information. If the export file does not exist, it is created, otherwise it is overwritten.

If you specify the optional argument UTF8 = True, the export file will be encoded in the UTF8 format (local character set independent format).

See also Record

MoveTheAsset (ToPathName As String)

This method moves the record's asset file to the given folder path and updates the record's asset asset reference.

You can specify only a Windows destination **folder name**, but not a new file name. If ToPathName is an empty string (""), a dialog will be prompted and you can select an asset storage (if necessary) and a target location.

See also Record

ShowPreviewWindow

This method shows the record’s asset file’s preview window in the application frame.

See also Record

ConvertAsset (ToFolder As String, [,Parameters As String] [,AssetStorageUID As String]) As String

This method uses a asset conversion Asset Storage Module to create a new asset from the given asset (associated with the record). The method returns the location (Windows file path) of the newly created asset.

The new asset will be created in the folder specified with the ToFolder argument. You have to specify a Windows folder path, not a file path.

If you omit the optional Parameters argument (or set it to ""), the Asset Storage Module will prompt a dialog showing several conversion options. If you pass a Parameters string, these values are used and no dialog is shown. You need specific development information to design this parameter string.

The argument AssetStorageUID (a GUID string value) allows you to select a specific Asset Storage Module. If you omit this argument or set it to "", the standard converter with the GUID UID_AS_PIXEL_CONVERTER is used as default.

It is permissible to specify the name of the Asset Storage Module instead of the UID (for example "Pixel Image Converter").

See also Record

OptimizeThumbnail

This method tries to optimize the record’s thumbnail.

Note: This method applies to the specific catalog field UID_REC_THUMBNAIL (see also FieldAccessConstants).

See also Record

RotateThumbnail (ByDegrees As Long)

This method rotates the thumbnail by the given amount of degrees. The parameter must be a multiple of 90.

Note: This method applies to the specific catalog field UID_REC_THUMBNAIL (see also FieldAccessConstants).

See also Record

ShowInformationWindow

This method shows the record's information window.

See also Record

ShowIPTCInformationWindow

This method shows the record's IPTC information window.

See also Record

Update ([Always As Boolean])

This method updates the record if necessary. If the optional parameter “Always” is set to “True,” it will be updated in every case.

See also Record

Save

This method saves the record’s changes permanently to the catalog's database. You have to use the method after calling these record field modification methods:

(set) Field.Value
Field.PastePicture
Field.PastePictureFromFile
(set) Record.Asset

See also Record

SavePreviewToFile (MaxWidth As Long, MaxHeight As Long, Path As String [,Rotation As Boolean])

This method creates a Windows bitmap file (*.bmp) from the record's asset preview (without showing the preview).

You can specify the MaxWidth and MaxHeight values for the created bitmap file (two values of 0 means original size). Additional you have to specify a target path.

Note: The arguments MaxWidth and MaxHeight can't be used for the scaling of every preview bitmap. This depends on the filter that is invoked.

If the optional parameter Rotation is set to **true**, the created bitmap file will be rotated corresponding to the thumbnail rotation state.

The arguments work for example for preview from
- EPS files / "Cumulus 5 EPS Filter"
- TIF files / "Quicktime Filter"

For other filters, the values are ignored, for example for previews from
- BMP files / "Cumulus 4 Compatibility Filter
- TIF files / "Cumulus 4 Compatibility Filter

VB sample code:

```
rec.SavePreviewToFile 0, 0, "c:\x1.bmp"
rec.SavePreviewToFile 400, 400, "c:\x2.bmp"
```

See also Record

Application As Application

This property returns the Application object.

See also Server

Parent As Object

This property returns the server’s parent object.

See also Server

IsAdministrator As Boolean

This property returns if the current connection to the server is an administrator’s connection.

See also Server

Name As String

This property returns the server’s name.

See also Server

Catalogs As ServerCatalogs

This property returns the server's collection of open catalogs.

See also Server

Application As Application

This property returns the Application object.

See also ServerCatalog

Parent As Object

This property returns the server catalog's parent object.

See also ServerCatalog

Name As String

This property returns a remote catalog’s name.

This is the **default** property.

See also ServerCatalog, Catalog.Name

ID As Long

This property returns a remote catalog's ID.

Note: This is NOT the ID returned from the Catalog.ID property.

See also ServerCatalog

IsShared As Boolean (Administrator only)

This property returns or sets a remote catalog’s sharing state.

See also ServerCatalog, Catalog.AllowSharing

IsPublishedToInternet As Boolean (Administrator only)

This property returns or sets if the catalog is published to the World Wide Web.

See also ServerCatalog, Catalog.PublishToWWW

Open As CatalogCollection

This method opens a remote catalog at the server and returns the remote catalog's CatalogCollection object.

Note: If this catalog is opened by another administrator, the other administrator will be disconnected from the catalog. This assures that only one administrator is connected to a catalog.

If you want to open several catalogs served by a Cumulus Server you have to establish a separate connection to the server first.

VB sample code:

```
Dim s1 As Server
Dim s2 As Server
Dim scats As ServerCatalogs
Dim scat As ServerCatalog
Dim scat2 As ServerCatalog

Set s1 = App.ConnectToServer("ServerA", "", "")
Set s2 = App.ConnectToServer("ServerB", "", "")

Set scats = s1.Catalogs
Set scat = scats.Item("Test")
scat.Open

Set scats = s2.Catalogs
Set scat2 = scats.Item("Test2")
scat2.Open

MsgBox scat
MsgBox scat2
```

See also ServerCatalog

Application As Application

This property returns the Application object.

See also ServerCatalogs

Count As Long

This property returns the number of open remote catalogs.

See also ServerCatalogs

Parent As Object

This property returns the server catalog collection’s parent object.

See also ServerCatalogs

Add (Pathname As String [, DatabaseSelectorUID As String]) As ServerCatalog (Administrator only)

This method adds an existing catalog to the server's catalog list using an optional write access password and returns the server catalog object.

The argument DatabaseSelectorUID (a GUID string value) allows you to select a specific database engine. If you omit this argument or set it to "", the Cumulus Engine is used as default. Some possible values are:

UID_SELECTOR_CUE (Cumulus Engine)
UID_SELECTOR_ODBC (ODBC Connector)

It is allowed to specify the name of the database selector instead of the UID (for example "Cumulus Database Engine" or "ODBC Database Connector").

See also ServerCatalogs, FieldAccessConstants

Item (Index As Long / Name As String) As ServerCatalog

This property returns a collection's server catalog by index or by name.

This is the **default** property.

See also ServerCatalogs

ItemByID (ID As Long) As ServerCatalog

This property returns a collection’s server catalog by the server catalog’s ID.

See also ServerCatalogs, ServerCatalog.ID

New (Pathname As String [,CatalogName As String] [,DatabaseSelectorUID As String]) As ServerCatalog (Administrator only)

This method creates a new catalog at the server, adds it to the server's catalog list and returns the server catalog object.

The catalog name is optional, the file name willl be taken as default if you omit the Catalog Name or set it to "" (necessary if you want to specify the DatabaseSelectorUID).

The argument DatabaseSelectorUID (a GUID string value) allows you to select a specific database engine. If you omit this argument or set it to "", the Cumulus Engine is used as default. Some possible values are:

UID_SELECTOR_CUE (Cumulus Engine)
UID_SELECTOR_ODBC (ODBC Connector)

It is allowed to specify the name of the database selector instead of the UID (for example "Cumulus Database Engine" or "ODBC Database Connector").

See also ServerCatalogs, FieldAccessConstants

Remove (Index As Long / Name As String) (Administrator only)

This property removes a server catalog from the collection by index or by name.

See also ServerCatalogs

RemoveByID (ID As Long) (Administrator only)

This property removes a server catalog from the collection by the catalog’s ID.

See also ServerCatalogs, ServerCatalog.ID

Restore (BackupPathName As String, PathName As String [,DatabaseSelectorUID As String]) As ServerCatalog (Administrator only)

This method restores a catalog backup file “BackupPathName” and creates a new catalog “PathName” at the server. The catalog is added to the server’s catalog list and is returned as a server catalog object.

The argument DatabaseSelectorUID (a GUID string value) allows you to select a specific database engine. If you omit this argument or set it to "", the Cumulus Engine is used as default. Some possible values are:

UID_SELECTOR_CUE (Cumulus Engine)
UID_SELECTOR_ODBC (ODBC Connector)

It is allowed to specify the name of the database selector instead of the UID (for example "Cumulus Database Engine" or "ODBC Database Connector").

See also ServerCatalogs, FieldAccessConstants

Application As Application

This property returns the Application object.

See also UserPermission

CanDeleteRecords As Boolean

This property returns or sets if the user is allowed to delete records in the catalog.

See also UserPermission

CanModifyCategories As Boolean

This property returns or sets if the user is allowed to modify the category list (adding, changing or deleting categories).

See also UserPermission

CanModifyRecords As Boolean

This property returns or sets if the user is allowed to modify records.

See also UserPermission

CanTransferAssets As Boolean

This property returns or sets if the user is allowed to transfer assets.

See also UserPermission

CanOpenCatalog As Boolean

This property returns or sets if the user is allowed to open the catalog.

See also UserPermission

Name As String

This property returns or sets the user’s name of the user permission.

This is the **default** property.

See also UserPermission

GroupID As Long (Enterprise Edition only)

This property returns or sets the user’s group id. The user is member of the group if he has the corresponding group ID (see Group.ID).

See also UserPermission

HTMLTemplate As String (Enterprise Edition only)

This property returns or sets the user’s HTMLTemplate file as string.

The statements in this HTML Template file define the HTML pages which can be accessed by a user when working with the Cumulus Internet Client.

See also UserPermission

RecordFilter As String (Enterprise Edition only)

This property returns or sets the user’s Record Filter query. This value describes a search query for records that limits the range of visible records for this user.

An empty string value ("") clears the current Record Filter and allows this user to see all (!) records.

Please see CatalogCollection.Query / CatalogCollection.Find for a description of the query string syntax.

See also UserPermission

CategoryFilter As String (Enterprise Edition only)

This property returns or sets the user’s Category Filter query. This value describes a search query for categories that limits the range of visible categories for this user.

An empty string value ("") clears the current Category Filter and allows this user to see all (!) categories.

Please see CatalogCollection.QueryCategories / CatalogCollection.FindCategories for a description of the query string syntax.

See also UserPermission

Parent As Object

This property returns the user permission's parent object.

See also UserPermission

Application As Application

This property returns the Application object.

See also UserPermissions

Count As Long

This property returns the number of user permissions.

See also UserPermissions

Parent As Object

This property returns the user permissions’ parent object.

See also UserPermissions

Add (Name As String) As UserPermission

This method adds a new user permission to the user permission list using the given name and returns the user permission object.

See also UserPermissions

Item (Index As Long / Name As String) As UserPermission

This property returns a user permission by index or by user name.

This is the **default** property.

See also UserPermissions

Remove (Index As Long / Name As String)

This method removes a user permission from the user permission list by index or by user name.

See also UserPermissions

Application As Application

This property returns the Application object.

See also Layout

Parent As Object

This property returns the Layout's parent object.

See also Layout

FieldDefinitions As FieldDefinitions

This property returns the FieldDefinitions collection object of the Layout. You can use it to enumerate all fields of the layout to obtain information on the fields contained.

This is the **default** property.

See also Layout

CreateNewFieldUID As String

This method creates a new FieldUID (this is a GUID in COM terminology). This FieldUID can be used to create a new field in the Layout using the FieldDefinitions.Add method.

An example of a valid FieldUID is "{5251a742-727c-11d2-a73d-0000c000cdd3}".

See also Layout

AvailableFields As LinkList

This property returns a list of all field entries (LinkList object) that can be used to set up the RecordLayout of a catalog.

This list describes fields that can be filled during cataloging (adding) or updating of records if the fields are part of the catalog's record layout.

The link list is not automatically part of the Record Layout, but can be used to obtain information on the fields that can be filled from the cataloger, filters and the application.

Note: This property is only available if the Layout object is taken from the Catalog.RecordLayout (not available for the CategoryLayout).

See also Layout

CreateValueList (ProposedName As String) As ValueList

This property creates a new ValueList object. The given name is just a user suggestion. The created name can be requested by using the "Name" method of the returned ValueList object.

See also Layout

RemoveValueList (ListName As String)

This method removes a temporarily created ValueList object. The List Name can be requested by using the "Name" method of a previously created ValueList object (using the CreateValueList method).

See also Layout

Add (FieldUID As String, Type As Long, DefaultName As String, [ValueList As ValueList])

This method adds a new field to the FieldDefinitions. It is defined by the field ID (a GUID string value), the type (see below), the default name (English) and the ValueList object (if type is FieldTypeEnum).

These constants describe the type of a catalog’s field. Possible values are:

FieldTypeString	= 0
FieldTypeBool	= 1
FieldTypeInteger	= 2
FieldTypeDouble	= 3
FieldTypeDate	= 4
FieldTypeBinary	= 5
FieldTypePicture	= 6
FieldTypeEnum	= 7
FieldTypeAudio	= 8

You can add an Integer type field like this:

```
Dim FieldUID As String
FieldUID = Layout.CreateNewFieldUID
Layout.FieldDefinitions.Add FieldUID, FieldTypeInteger, "NewIntField"
```

In case of Type = FieldTypeEnum, the field can be added like this:

```
Private Sub Test(Lay As Layout)
    Dim vl As ValueList
    Set vl = Lay.CreateValueList("Colors")
    vl.Add "Blue"
    vl.Add "Red"
    Layout.FieldDefinitions.AddField Lay.CreateNewFieldUID, FieldTypeEnum, "NEW_1", vl
End Sub
```

Please see the corresponding section of Field.Value for a VB sample code for getting / setting FieldTypeEnum record field data.

See also FieldDefinitions, LayoutFieldTypeConstants and ValueList

Application As Application

This property returns the Application object.

See also FieldDefinitions

ListOfData (What As Long) As Variant

This read-only property returns a list of information taken from all (!) FieldDefinition entries. You can use this function to get the information faster and more efficiently than calling the FieldDefinitions.Item function in a loop to request single FieldDefinition objects and call their properties. The returned array contains the information in the same order as if you looped over the single items by index. The array corresponds to the list returned from Fields.ListOfData.

The "What" argument specifies what kind of information is returned and can have one of the following values:

ListOfFieldType = 1
Returns an array of Long (equivalent to FieldDefinition.FieldType).

ListOfFieldUID = 2
Returns an array of String (equivalent to FieldDefinition.FieldUID).

ListOfFieldName = 3
Returns an array of String (equivalent to FieldDefinition.Name).

ListOfShortID = 4
Returns an array of Long (equivalent to FieldDefinition.ShortID).

ListOfAllowEdit = 7
Returns an array of Boolean (equivalent to FieldDefinition.AllowUserToEdit).

VB sample code:

```
Dim Defs As FieldDefinitions
Set Defs = Layout.FieldDefinitions ' Layout taken from anywhere

Dim Types As Variant
Dim IDs As Variant
Dim Names As Variant
Dim ShortIDs As Variant
Dim AllowEdit As Variant

Names = Defs.ListOfData(ListOfFieldName)
IDs = Defs.ListOfData(ListOfFieldUID)
ShortIDs = Defs.ListOfData(ListOfShortID)
Types = Defs.ListOfData(ListOfFieldType)
AllowEdit = Defs.ListOfData(ListOfAllowEdit)

size = Defs.Count

For i = 0 To size - 1
    MsgBox Names(i) & Chr(13) & _
            IDs(i) & Chr(13) & _
            ShortIDs(i) & Chr(13) & _
            Types(i) & Chr(13) & _
            AllowEdit(i)
Next
```

See also FieldDefinitions, ListOfDataConstants

Count As Long

This property returns the number of fields in the Layout (FieldDefinitions collection).

See also FieldDefinitions

Parent As Object

This property returns the FieldDefinition’s parent object.

See also FieldDefinitions

Remove (Index As Long / Name As String)

This method removes a field from the Layout (FieldDefinitions collection). RemoveField can be used by Index (of the collection) or by Name (of the Field).

VB sample code:

```
Private Sub Test_Layout(L As Layout)
    Dim defs As FieldDefinitions
    Set defs = L.FieldDefinitions
    MsgBox defs.Count ' will show n
    defs.Remove 0 ' removes the first of all fields in the collection
    MsgBox defs.Count ' will show n-1
End Sub
```

Note: it is not allowed to remove the following basic **category fields**:

UID_CAT_NAME
UID_CAT_WIN_FOLDER_PATH
UID_CAT_URL_FOLDER_PATH
UID_CAT_MAC_VOLUME_ID
UID_CAT_MAC_DIRECTORY_ID
UID_CAT_TYPE

See also FieldDefinitions

RemoveByID (ShortID As Long / FieldUID As String)

This method removes a field from the Layout (FieldDefinitions collection). RemoveField can be used by ShortID (a Long value) or by FieldUID (a GUID string value).

The different predefined field id constants are listed in FieldAccessConstants.

VB sample code:

```
Private Sub Test_Layout(L As Layout)
    Dim defs As FieldDefinitions
    Set defs = L.FieldDefinitions
    MsgBox defs.Count' will show n
    defs.RemoveByID UID_REC_HEIGHT ' removes this field
    MsgBox defs.Count ' will show n-1
End Sub
```

Note: it is not allowed to remove the following basic **category fields**:

```
    UID_CAT_NAME
    UID_CAT_WIN_FOLDER_PATH
    UID_CAT_URL_FOLDER_PATH
    UID_CAT_MAC_VOLUME_ID
    UID_CAT_MAC_DIRECTORY_ID
    UID_CAT_TYPE
```

See also FieldDefinitions

Item (Index As Long / Name As String) As FieldDefinition

This method gets a FieldDefinition object from the Layout (FieldDefinitions collection). Item can be used by Index (of the collection) or by Name (of the Field).

This is the **default** property.

Note: ItemByID used with the FieldUID (GUID string) is the fastest way of access. Item (Name) is slower.

VB sample code:

```
Private Sub Test_Layout(L As Layout)
    Dim defs As FieldDefinitions
    Dim def As FieldDefinition

    Set defs = L.FieldDefinitions

    Set def = defs.Item("Record Name")
    MsgBox def.Name(LanguageUSEnglish) ' will display "Record Name"

    Set def = defs.Item(0)
    MsgBox def.Name(LanguageUSEnglish) ' will display the name of the first field
End Sub
```

Note: This method does not raise an error in case of failure. It simply returns "Nothing." Hence, you can use Item to check if a field exist in the layout

```
    If defs.Item("Field XYZ") Is Nothing Then
        MsgBox "Field not available!"
    End If
```

See also FieldDefinitions

ItemByID (ShortID As Long / FieldUID As String) As FieldDefinition

This method gets a FieldDefinition object from the Layout (FieldDefinitions collection). ItemByID can be used by ShortID (a Long value) or by FieldUID (a GUID string value).

ItemByID used with the FieldUID (GUID string) is the fastest way of access (faster than Item).

The different predefined field id constants are listed in FieldAccessConstants.

VB sample code:

```
Private Sub Test_Layout(L As Layout)
    Dim defs As FieldDefinitions
    Dim def As FieldDefinition

    Set defs = L.FieldDefinitions

    Set def = defs.ItemByID(UID_REC_HEIGHT) ' Request uses the GUID string
    MsgBox def.Name(LanguageUSEnglish) ' will display "Image Height"

    Dim S_ID As Long
    S_ID = def.shortID
    MsgBox S_ID ' will display 1214605671
    Set def = defs.ItemByID(S_ID) ' Request uses the internal short ID
    MsgBox def.Name(LanguageUSEnglish) ' will display "Image Height"
End Sub
```

Note: This method does not raise an error in case of failure. It simply returns "Nothing." Hence, you can use ItemByID to check whether a field exists in the layout

```
    If defs.ItemByID(UID_REC_COLOR_MODE) Is Nothing Then
        MsgBox "Field not available!"
    End If
```

See also FieldDefinitions

Application As Application

This property returns the Application object.

See also FieldDefinition

Parent As Object

This property returns the FieldDefinition’s parent object.

See also FieldDefinition

AllowUserToEdit As Boolean

This property returns or sets the "Allow User To Edit" state of the field.

See also FieldDefinition

UseInitialValue As Boolean

This property returns or sets the "Use Initial Value" state of the field.

If this switch is set and no other field linking is active for this field, the value defined under InitialValue is taken when adding and/or updating records.

Note: If the FieldDefinition object is taken from the Catalog.CategoryLayout, this property is NOT available for the following basic category fields:

UID_CAT_WIN_FOLDER_PATH
UID_CAT_URL_FOLDER_PATH
UID_CAT_MAC_VOLUME_ID
UID_CAT_MAC_DIRECTORY_ID
UID_CAT_TYPE
UID_CAT_ORIGINAL_ID
UID_CAT_PARENT_ID

See also FieldDefinition

InitialValue As String

This property returns or sets the "Initial Value" of the field.

If the switch UseInitialValue is set and no other field linking is active for this field, the InitialeValue is taken when cataloging (adding) and/or updating records.

Note: If the FieldDefinition object is taken from the Catalog.CategoryLayout, this property is NOT available for the following basic category fields:

UID_CAT_WIN_FOLDER_PATH
UID_CAT_URL_FOLDER_PATH
UID_CAT_MAC_VOLUME_ID
UID_CAT_MAC_DIRECTORY_ID
UID_CAT_TYPE
UID_CAT_ORIGINAL_ID
UID_CAT_PARENT_ID

See also FieldDefinition

LinkToFields As LinkList

This property returns the field's LinkList object.

This object contains all fields that are used as a source for the field linking during the cataloging (adding) or updating of records if the switch LinkToAllMatchingFields is set to false (which means: link to these fields).

Note: This property is only available if the FieldDefinition object is taken from the Catalog.RecordLayout (available for the CategoryLayout in the Enterprise Edition only).

See also FieldDefinition

LinkWhenAdding As Boolean

This property returns or sets the "Linking when Adding Assets" state of the field.

Note: This property is only available if the FieldDefinition object is taken from the Catalog.RecordLayout (available for the CategoryLayout in the Enterprise Edition only).

See also FieldDefinition

LinkWhenUpdating As Boolean

This property returns or sets the "Linking when Updating Assets" state of the field.

Note: This property is only available if the FieldDefinition object is taken from the Catalog.RecordLayout (available for the CategoryLayout in the Enterprise Edition only).

See also FieldDefinition

LinkWhenSavingToAsset As Boolean

This property returns or sets the "Linking when Saving to Assets" state of the field.

Note: This property is only available if the FieldDefinition object is taken from the Catalog.RecordLayout (available for the CategoryLayout in the Enterprise Edition only).

See also FieldDefinition

ReadAllWhenAdding As Boolean

This property returns or sets the "When Adding Assets Read All" state of the field. If the switch LinkWhenAdding is set, the ReadAllWhenAdding determines whether the field is filled from all linked fields, or only from the first one.

Note: This property is only available if the FieldDefinition object is taken from the Catalog.RecordLayout (available for the CategoryLayout in the Enterprise Edition only).

See also FieldDefinition

ReadAllWhenUpdating As Boolean

This property returns or sets the "When Updating Assets Read All" state of the field. If the switch LinkWhenUpdating is set, the ReadAllWhenUpdating determines whether the field is filled from all linked fields, or only from the first one.

Note: This property is only available if the FieldDefinition object is taken from the Catalog.RecordLayout (available for the CategoryLayout in the Enterprise Edition only).

See also FieldDefinition

LinkToAllMatchingFields As Boolean

This property returns or sets the "Link to All Matching Fields" state of the field.

If this switch is set to false, the property LinkToFields defines a list of user-specified fields for the field linking.

Note: This property is only available if the FieldDefinition object is taken from the Catalog.RecordLayout (available for the CategoryLayout in the Enterprise Edition only).

See also FieldDefinition

FieldUID As String

This property returns the FieldUID (as a GUID string) of this field (represented by the FieldDefinition object).

This is the **default** property.

The FieldUID is a GUID in string representation such as "{5251a742-727c-11d2-a73d-0000c000cdd3}".

See also FieldDefinition

ShortID As Long

This property returns the ShortID (as a Long Value) of this field (represented by the FieldDefinition object). This Long value is used internally in Cumulus to access fields. This value can be used alternatively to the FieldUID (as GUID string) to access fields of a layout.

See also FieldDefinition

ValueList As ValueList

This property returns the ValueList object of this field (represented by the FieldDefinition object).

A field has a ValueList if the field type is FieldTypeEnum.

VB sample code:

```
Private Sub Test_Layout(L As Layout)
    Dim defs As FieldDefinitions
    Dim def As FieldDefinition
    Dim vl As ValueList

    Set defs = L.FieldDefinitions
    Set def = defs.ItemByID(UID_REC_COLOR_MODE)
    Set vl = def.ValueList

    MsgBox vl(0)
    MsgBox vl(1)
    MsgBox vl(2)
End Sub
```

Please see the corresponding section of Field.Value for a VB sample code for getting / setting FieldTypeEnum record field data.

See also FieldDefinition

FieldType As Long

This property returns the type of this field (represented by the FieldDefinition object).

The different field type constants are listed in LayoutFieldTypeConstants.

Possible values are:

FieldTypeString	= 0
FieldTypeBool	= 1
FieldTypeInteger	= 2
FieldTypeDouble	= 3
FieldTypeDate	= 4
FieldTypeBinary	= 5
FieldTypePicture	= 6
FieldTypeEnum	= 7
FieldTypeAudio	= 8

See also FieldDefinition

Name ([LanguageID As Long]) As String

This property returns or sets the name of this field (represented by the FieldDefinition object).

If you do not specify a language ID, the current Application.LanguageID value is taken.

```
Dim s As String
Dim Field as FieldDefinition
Set Field = Layout.FieldDefinitions.ItemByID(UID_REC_RECORD_NAME) ' Layout taken from Catalog
s = Field.Name
```

See also FieldDefinition, LanguageConstants

IndexKind As Long

This property returns or sets the specified field’s indexing state.

Possible values are:

IndexKindNotIndexed	= 0
IndexKindSortIndexed	= 1
IndexKindContentIndexed	= 2

These values may be combined by adding. A Value of 3 means IndexKindSortIndexed and IndexKindContentIndexed.

Note: This property is only available if the FieldDefinition object is taken from the Catalog.RecordLayout (available for the CategoryLayout in the Enterprise Edition only).

Note: If the FieldDefinition object is taken from the Catalog.CategoryLayout, (Enterprise Edition only) the setting of the IndexKind property is NOT available for the following basic category fields:

UID_CAT_NAME
UID_CAT_WIN_FOLDER_PATH
UID_CAT_URL_FOLDER_PATH
UID_CAT_MAC_VOLUME_ID
UID_CAT_MAC_DIRECTORY_ID
UID_CAT_TYPE
UID_CAT_ORIGINAL_ID
UID_CAT_PARENT_ID

See also FieldDefinition, IndexKindTypeConstants.

Application As Application

This property returns the Application object.

See also ValueList

ID As Long

This property returns the Value List's unique ID. This ID is valid until the catalog is closed. If the same catalog is reopened, this ValueList it will have another ID.

See also ValueList

Count As Long

This property returns the number of strings of the ValueList.

See also ValueList

Parent As Object

This property returns the ValueList’s parent object.

See also ValueList

Name As String

This property returns the name of the ValueList object.

See also ValueList

Add (Value As String [, BeforeIndex As Long]) As Long

This method adds a new string to the ValueList. The optional parameter BeforeIndex defines the destination index. If no index is given, the string is appended to the list.

The return value contains the ID of the new entry (see also ValueList.AddByID).

The language of the name is the current value of Application.LanguageID.

To assign different language values to this entry, you have to get a ValueListEntry object for each language for which you want to get/set values. You can use ValueList.Item or ValueList.ItemByID.

If you use the default property mechanism, you can assign a language like this:

```
ValueList.Item(0, LanguageGerman) = "German text"
ValueList.Item(0, LanguageJapanese) = "Japanese text"
```

See also ValueList and LanguageConstants

AddByID (Value As String, ID As Long [, BeforeIndex As Long])

This method adds a new string to the ValueList. The optional parameter BeforeIndex defines the destination index. If no index is given, the string is appended to the list.

The parameter ID will be used to create a new entry with this ID (see also ValueList.Add). Take care to have unique IDs within your Value List. You may not have two string entries with the same ID.

The language of the name is the current value of Application.LanguageID.

To assign different language values to this entry, you have to get a ValueListEntry object for each language for which you want to get/set values. You can use ValueList.Item or ValueList.ItemByID.

If you use the default property mechanism, you can assign a language like this:

```
ValueList.Item(0, LanguageGerman) = "German text"
ValueList.Item(0, LanguageJapanese) = "Japanese text"
```

See also ValueList and LanguageConstants

Item (Index As Long [, LanguageID As Long]) As ValueListEntry

This property returns a ValueListEntry object from the ValueList by index. If you do not specify a language ID, the current Application.LanguageID value is taken.

This is the **default** property.

Please see the corresponding section of Field.Value for a VB sample code for getting / setting FieldTypeEnum record field data.

See also ValueList and LanguageConstants

ItemByID (ID As Long [, LanguageID As Long]) As ValueListEntry

This property returns a ValueListEntry object from the ValueList by ID. If you do not specify a language ID, the current Application.LanguageID value is taken.

Please see the corresponding section of Field.Value for a VB sample code for getting / setting FieldTypeEnum record field data.

See also ValueList and LanguageConstants

Remove (Index As Long, ReplacementID As Long) (Administrator only)

This method removes an entry from the ValueList specified by a given index.

The ReplacementID will be used to overwrite all fields in the corresponding catalog if the catalog fields are referring to this ValueList and the field entry is equal to ID. In this case, the substitution is done with the ReplacementID.

See also ValueList

RemoveByID (ID As Long, ReplacementID As Long) (Administrator only)

This method removes an entry from the ValueList specified by a given ID.

The ReplacementID will be used to overwrite all fields in the corresponding catalog if the catalog fields are referring to this ValueList and the field entry is equal to ID. In this case, the substitution is done with the ReplacementID.

See also ValueList

Application As Application

This property returns the Application object.

See also ValueListEntry.

Parent As Object

This property returns the ValueListEntry’s parent object.

See also ValueListEntry.

ID As Long

This read-only property returns the ID value of this ValueListEntry object.

See also ValueListEntry.

LanguageID As Long

This read-only property returns the language that is used for this ValueListEntry object.

The value is one of the LanguageConstants.

See also ValueListEntry.

Value As String

This property allows to get and set (String) values for this entry of the ValueList.

This is the **default** property.

To assign different language values to this entry, you have to get a ValueListEntry object for each language for which you want to get/set values. You can use ValueList.Item or ValueList.ItemByID.

If you use the default property mechanism, you can assign a language like this:

```
ValueList.Item(0, LanguageGerman) = "German text"
ValueList.Item(0, LanguageJapanese) = "Japanese text"
```

Please see the corresponding section of Field.Value for a VB sample code for getting / setting FieldTypeEnum record field data.

See also ValueListEntry.

Application Object

This is the Cumulus Application object.

- Activate
- Application
- CatalogCollections
- CatalogingOptions
- ConnectToServer
- ConvertToUTF16
- ConvertToUTF8
- CopyAsset
- CountContainedAssets
- DeleteAsset
- FindToken
- FrontCollection
- Fullname
- GenericEvent
- GetContainedAsset
- LanguageID
- ListContainedAssetNames
- Name
- NewCatalog
- OpenCatalog
- OpenCatalogByReference
- PaletteMode
- Parent
- Preferences
- Quit
- RestoreCatalog
- RestoreFactoryDefaults
- UsedApplicationList
- Visible

See also Overview

Catalog Object

The catalog object allows you to access an open Cumulus catalog.

- AllowSharing
- Application
- AssetInformationConfiguration
- Backup
- CanDeleteRecords
- CanModifyCategories
- CanModifyRecords
- CanTransferAssets
- CatalogReference
- CatalogServer
- CategoryInformationConfiguration
- CategoryLayout
- CentralAssetLocation
- CentralAssetLocationUsage
- CentralAssetLocationXML
- Compress
- ConvertTo
- FindAssetXML
- FullName
- Groups
- ID
- IsAdmin
- IsLocked
- IsReadOnly
- IsRemoteCatalog
- KeepCategoryNamesUnique
- KeepDeleted
- Name
- NativeThumbnailSize
- NewEmptyCopy
- Parent
- PublishToWWW
- Rebuild
- RecordLayout
- RecordsDeleted
- RecoverDeletedRecords
- RestrictedPermissions
- SendMessage
- SetAsDefault
- Unlock
- UserName
- UserPermissions
- WriteAccessPassword
- WriteMetadataToAssets

See also Overview

Records Object

The Records object is a collection containing record objects. The content of this list depends on the origin of the Record object (two sources):

CatalogCollection.Records: The collection represents all records in the collection window (record pane of the view).

CatalogCollection.SelectedRecords: All records that are selected in the record pane of the Collection.

Add
Application
CopyTheAssets
Count
Export
ExportToHTML
Item
ItemByID
MailTheAssets
MoveTheAssets
New
Parent
Remove
RemoveByID

See also Overview

Record Object

The Record Object represents a single record of a Cumulus catalog and allows access to the individual record members.

- Application
- Asset
- AssetReferences
- AssociatedProgram
- Categories
- CheckIn
- CheckOut
- CollectionIndex
- ConvertAsset
- CopyTheAsset
- Delete
- EditTheAsset
- Export
- Fields
- ID
- IsReadOnly
- MailTheAsset
- MoveTheAsset
- OptimizeThumbnail
- Parent
- PrintTheAsset
- RotateThumbnail
- Save
- SavePreviewToFile
- ShowLocation
- ShowPreviewWindow
- ShowInformationWindow
- ShowIPTCInformationWindow
- UndoCheckOut
- Update
- UpdateAssetReference

See also Overview

Server Object

The server object allows you to connect to a remote server and administer or work with it.

- Application
- Catalogs
- IsAdministrator
- Name
- Parent

See also Overview

ServerCatalog Object

The ServerCatalog object allows you to access an open remote Cumulus catalog.

Application
ID
IsShared
IsPublishedToInternet
Name
Open
Parent

See also Overview

ServerCatalogs Object

The ServerCatalogs object is a collection containing ServerCatalog objects.

Add
Application
Count
Item
ItemByID
New
Parent
Remove
RemoveByID
Restore

See also Overview

Categories Object

The Categories object is a collection containing several categories as a list sorted by name. The content of this list depends on the origin of the Categories object (five sources):

CatalogCollection.Categories: All Categories in the CatalogCollection.

CatalogCollection.CategoryTree: All Categories on the top level (root) of the CatalogCollection.

CatalogCollection.SelectedCategories: All categories that are selected in the category pane of the CatalogCollection.

Category.SubCategories: All subordinate categories of a category.

Record.Categories: All categories that are assigned to the record in question.

Add
AddRecordCategory
Application
CategoryNames
Count
Export
ExportVisiblesOnly
Item
ItemByID
New
Parent
Remove
RemoveByID

See also Overview

Category Object

The Category object allows you to access a single Category of a Cumulus catalog.

- Application
- Collapsed
- CategoryLayout
- CollectionIndex
- Export
- Fields
- ID
- IsReadOnly
- Level
- Original
- Parent
- ParentCategory
- ShowInformationWindow
- SubCategories

See also Overview

AssetReferences Object

The AssetReferences object allows you to access the individual asset references for different platforms of a record.

Application
Count
Item
Parent
Update
XML

See also Overview

AssetReference Object

The AssetReference Object allows you to access a single asset reference for a specific platform.

Application
AssetStorageUID
BinaryData
Level
Name
Parent
Path

See also Overview

CatalogingOptions Object

The CatalogingOptions object allows you to access the cataloging options of Cumulus.

- AddDuplicateAssets
- Application
- AssetFormatTable
- AssetStorages
- CatalogAllFormats
- CatalogingAction
- ColorThumbnailQuality
- CopyAssetToCentralLocation
- ExcludeDamagedAssets
- ExcludedAssets
- GrayscaleThumbnailQuality
- LineartThumbnailQuality
- OpenPropertyWindow
- OptimizeThumbnailAutomatically
- Parent
- ResolveShortcuts
- ShowProgressBar
- ThumbnailSource

See also Overview

Preferences

The Preferences object allows you to access the application preferences of Cumulus.

- Application
- ManualPreviewSize
- Parent
- PreviewSize
- ResolveRelatedCategories
- SearchAllSelectedCategories
- SearchForCategoriesAbove
- SearchForCategoriesBelow
- ShowWarnings
- ThumbnailSize
- ThumbnailSimilarityThreshold
- Units
- UseGrid
- WindowToOpenOnDoubleClick

See also Overview

UserPermission Object

The UserPermission object allows you to access a single user permission of the catalog.

Application
CanDeleteRecords
CanModifyCategories
CanModifyRecords
CanOpenCatalog
CanTransferAssets
CategoryFilter
GroupID
HTMLTemplate
Name
Parent
RecordFilter

See also Overview

UserPermissions Object

The UserPermissions object allows you to access the user permissions of a catalog.

Add
Application
Count
Item
Parent
Remove

See also Overview

AssetFormatEntry Object

The AssetFormatEntry object allows you to access a single asset format entry.

Application
Extension
FileFormat
IsActive
Name
Parent

See also Overview

AssetFormatTable Object

The AssetFormatTable object allows you to access the asset format entries.

Application
Count
Item
Parent
MoveAssetFormatEntry

See also Overview

CatalogCollections Object

The CatalogCollections object allows you to access all collections. This is the list of all open catalog windows showing any views of any catalogs.

- Add
- Application
- Count
- Item
- ItemByID
- Parent
- Remove
- RemoveByID

See also Overview

CatalogCollection Object

The collection represents one catalog window (view). Cumulus allows to open several independent views of the same catalog.

- Add
- AddByID
- Application
- Catalog
- CatalogAsset
- Categories
- CategoryListVisible
- CategoryTree
- DetailsViewConfiguration
- Find
- FindAll
- FindAllCategories
- FindCategories
- FindMatchingRecords
- ID
- ImportCategories
- ImportRecords
- Mail
- MoveRecord
- MoveRecordByID
- Name
- New
- PaletteViewConfiguration
- Parent
- Query
- QueryCategories
- RecordListVisible
- Records
- Remove
- RemoveByID
- RemoveCategory
- RemoveCategoryByID
- Save
- SelectedCategories
- SelectedRecords
- SortDirection
- Sorting
- SortingByID
- ThumbnailViewConfiguration
- View
- WindowHandle

See also Overview

Layout Object

The Layout specifies the number, type and names of the catalog’s fields.

- Application
- AvailableFields
- CreateNewFieldUID
- CreateValueList
- FieldDefinitions
- Parent
- RemoveValueList

See also Overview

Field Object

The Field object allows you to access the value of a single field of a record or a category of the catalog.

- Application
- CopyPicture
- CopyPictureToFile
- FieldUID
- HasValue
- Parent
- PastePicture
- PastePictureFromFile
- Value

See also Overview

Fields Object

The Field object allows you to access all fields of a record or of a category in the catalog.

Application
Count
Item
ItemByID
ListOfData
Parent

See also Overview

LinkList Object

The meaning of this object depends on the origin of the LinkList object.

FieldDefinition.LinkToFields:
The object describes the field linking of the field referred by the FieldDefinition object

Layout.AvailableFields:
The object describes fields that can be filled during cataloging (adding) or updating of records if the fields are part of the catalog's record layout. The link list is not automatically part of the Record Layout, but can be used to obtain information about the fields that can be filled by the cataloger, filters and the application.

- Add
- AddAllMatchingModuleFields
- Application
- Count
- Item
- MoveLinkEntry
- Parent
- Remove

See also Overview

LinkEntry Object

The link entry object represents an entry of the LinkList object. It contains information about the origin of this link entry (ModuleClass and FieldUID).

- Application
- FieldType
- FieldUID
- FieldValueList
- ModuleClassUID
- ModuleName
- Name
- Parent

See also Overview

ValueList Object

The ValueList object allows the handling of value lists inside the catalog. These lists are referred by catalog fields of the type FieldTypeEnum. A ValueList object can be obtained by calling FieldDefinition.ValueList.

- Add
- AddByID
- Application
- Count
- ID
- Item
- ItemByID
- Name
- Parent
- Remove
- RemoveByID

Please see the corresponding section of Field.Value for a VB sample code for getting / setting FieldTypeEnum record field data.

See also Overview

ValueListEntry Object

This object allows the handling of the individual ValueList elements.

Application
ID
LanguageID
Parent
Value

See also Overview

FieldDefinitions Object

This object can be used to enumerate all fields of the layout to obtain information about the fields contained.

- Add
- Application
- Count
- Item
- ItemByID
- ListOfData
- Parent
- Remove
- RemoveByID

See also Overview

FieldDefinition Object

This object can be used it to access a specific field of a layout.

- AllowUserToEdit
- Application
- FieldType
- FieldUID
- IndexKind
- InitialValue
- LinkToAllMatchingFields
- LinkToFields
- LinkWhenAdding
- LinkWhenSavingToAsset
- LinkWhenUpdating
- Name
- Parent
- ReadAllWhenAdding
- ReadAllWhenUpdating
- ShortID
- UseInitialValue
- ValueList

Note: Some of the FieldDefinition properties (field linking relevant) are only available if the FieldDefinition object is taken from Catalog.RecordLayout.FieldDefinitions.Item() (not available for the CategoryLayout).

See also Overview

ViewConfiguration Object

This collection object can be used to configure the properties of a CatalogCollection view. The meaning of this object depends on the origin of the ViewConfiguration object.

CatalogCollection.DetailsViewConfiguration:
The view configuration object describes the properties of the collection's details view.

CatalogCollection.ThumbnailViewConfiguration:
The view configuration object describes the properties of the collection's thumbnail view.

- Add
- Application
- BackgroundColor
- Count
- Item
- ItemByID
- MoveViewEntry
- Parent
- Remove
- RemoveByID
- RestoreDefaults
- SaveAsDefaults
- ShowFrame
- ThumbnailImageSize
- ThumbnailSpacing

See also Overview

InfoEntry Object

The meaning of this object depends on the origin of the InfoEntry object.

Catalog.AssetInformationConfiguration.Item:
The info entry object describes the properties of a shown field in the asset information window.

Catalog.CategoryInformationConfiguration.Item:
The info entry object describes the properties of a shown field in the category information window.

- Application
- Bold
- CharSet
- Color
- FieldUID
- FontName
- FontSize
- Italic
- Name
- NumLines
- Parent
- PictureSize
- StrikeOut
- Underline

See also Overview

Application As Application

This property returns the Application object.

See also InfoEntry

Bold As Boolean

This property returns or sets the **Bold** state of the font for this information field entry.

See also InfoEntry

CharSet As Long

This property returns or sets the CharSet state of the font for this field entry. The value can be one of the CharSetConstants and can be used to define the language character set of the font for this information field entry.

See also InfoEntry, CharSetConstants

Color As Long

This property returns or sets the Color of the font for this information field entry.

See also InfoEntry

FieldUID As String

This property returns the FieldUID (a GUID string value) of this information field entry.

You can use the FieldUID to request a FieldDefinition object to obtain more information about the field properties in the catalog.

VB sample code:

```
Dim Def As FieldDefinition
Set Def = Collection.Catalog.RecordLayout.FieldDefinitions.ItemByID(ViewEntry.FieldUID)
```

See also InfoEntry

FontName As String

This property returns or sets the name of the font for this information field entry.

See also InfoEntry

FontSize As Long

This property returns or sets the size (in point) of the font for this information field entry.

See also InfoEntry

Italic As Boolean

This property returns or sets the *Italic* state of the font for this information field entry.

See also InfoEntry

Name As String

This read-only property returns the name of the field represented by the information field entry.

This is the **default** property.

See also InfoEntry

NumLines As Long

This property returns or sets the number of lines for string fields (extension of the edit text box) for this information field entry.

The allowed range covers 1-255 lines.

See also InfoEntry

Parent As Object

This property returns the information field entry's parent object.

See also InfoEntry

PictureSize As Long

This   property returns or sets the picture size (in pixels, only for picture / thumbnail fields) for this information field entry.

The allowed range covers 16-1024 pixels.

See also InfoEntry

StrikeOut As Boolean

This property returns or sets the ~~StrikeOut~~ state of the font for this information field entry.

See also InfoEntry

Underline As Boolean

This property returns or sets the Underline state of the font for this information field entry.

See also InfoEntry

InformationConfiguration Object

This collection object can be used to configure the properties of an asset or category information window. The meaning of this object depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
The info window configuration object describes the properties of the asset information window belonging to this catalog.

Catalog.CategoryInformationConfiguration:
The info window configuration object describes the properties of the category information window belonging to this catalog.

- Add
- Application
- ColumnWidth
- Count
- Item
- ItemByID
- KeepWindowOnTop
- MoveInfoEntry
- Parent
- PlaySound
- Remove
- RemoveByID
- ShowAllFields
- ShowAllObjectsInOneWindow
- SynchronizeToSelection

See also Overview

Add (ShortID As Long / FieldUID As String / Name As String[, AsIndex As Long]) As InfoEntry

This method adds a new displayed field to the information window and returns the new InfoEntry object. The field must be a valid field from the RecordLayout / CategoryLayout and can be specified as ShortID (a Long value) or by FieldUID (a GUID string value) or the field name. You can specify an optional AsIndex value to determine the position of the new entry. If you do not specify the AsIndex value, the new field as appended at the end of the list of displayed record / category fields.

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
The method inserts a new field in the asset information window of the catalog.

Catalog.CategoryInformationConfiguration:
The method inserts a new field in the category information window of the catalog.

See also InformationConfiguration

Application As Application

This property returns the Application object.

See also InformationConfiguration

ColumnWidth As Long

This property returns or sets the column width of the information window. This is the width of the displayed data field column.

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
This property returns or sets the column width of the asset information window of the catalog.

Catalog.CategoryInformationConfiguration:
This property returns or sets the column width of the category information window of the catalog.

See also InformationConfiguration

Count As Long

This property returns the number of InfoEntry entries in the information window field list.

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
Count returns the number of field entries in the asset information window of the catalog.

Catalog.CategoryInformationConfiguration:
Count returns the number of field entries in the category information window of the catalog.

See also InformationConfiguration

Item (Index As Long / Name As String) As InfoEntry

This method gets a specific InfoEntry object from the InformationConfiguration. Item can be used by Index (of the field list) or by Name (of the Field).

This is the **default** property.

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
The range of index / name covers the complete set of shown fields in the asset information window of the catalog.

Catalog.CategoryInformationConfiguration:
The range of index / name covers the complete set of shown fields in the category information window of the catalog.

See also InformationConfiguration

ItemByID(ShortID As Long / FieldUID As String) As InfoEntry

This method gets a specific InfoEntry object from the InformationConfiguration. ItemByID can be used by ShortID (a Long value) or by FieldUID (a GUID string value).

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
The range of valid IDs covers the complete set of shown fields in the asset information window of the catalog.

Catalog.CategoryInformationConfiguration:
The range of valid IDs covers the complete set of shown fields in the category information window of the catalog.

See also InformationConfiguration

KeepWindowOnTop As Boolean

This property returns or sets the window position state of the information window. "Keep Window On Top" keeps the information window always over other collection windows.

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
This property returns or sets the "Keep Window On Top" state of the asset information window of the catalog.

Catalog.CategoryInformationConfiguration:
This property returns or sets the "Keep Window On Top" state of the category information window of the catalog.

See also InformationConfiguration

MoveInfoEntry (FromIndex As Long, ToIndex As Long)

This methods moves the specified InfoEntry object (FromIndex) within the InformationConfiguration field list. The target position is specified by the ToIndex argument.

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
The method moves a field in the catalog's asset information window to the target position.

Catalog.CategoryInformationConfiguration:
The method moves a field in the catalog's category information window to the target position.

See also InformationConfiguration

Parent As Object

This property returns the InformationConfiguration's parent object.

See also InformationConfiguration

PlaySound As Boolean

This property returns or sets the "PlaySound" value of the information window. "Play Sound " plays automatically the first audio field (if available) when the next asset or category data set is shown.

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
This property returns or sets the "Play Sound" state of the asset information window of the catalog.

Catalog.CategoryInformationConfiguration:
This property returns or sets the "Play Sound" state of the category information window of the catalog.

See also InformationConfiguration

Remove (Index As Long / Name As String)

This method removes a specific InfoEntry object from the InformationConfiguration field list. Remove can be used by Index (of the list) or by Name (of the Field).

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
The range of index / name covers the complete set of shown fields in the asset information window of the catalog. The specified field is removed from the information window field list.

Catalog.CategoryInformationConfiguration:
The range of index / name covers the complete set of shown fields in the category information window of the catalog. The specified field is removed from the information window field list.

See also InformationConfiguration

RemoveByID(ShortID As Long / FieldUID As String)

This method removes a specific InfoEntry object from the InformationConfiguration field list. RemoveByID can be used by ShortID (a Long value) or by FieldUID (a GUID string value).

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
The range of valid IDs covers the complete set of shown fields in the asset information window of the catalog. The specified field is removed from the information window field list.

Catalog.CategoryInformationConfiguration:
The range of valid IDs covers the complete set of shown fields in the category information window of the catalog. The specified field is removed from the information window field list.

See also InformationConfiguration

ShowAllFields As Boolean

This property returns or sets the "Show All Fields" value of the information window. "Show All Fields" controls wether all fields are displayed in the information window (or only the fields listed in the field list).

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
This property returns or sets the "Show All Fields" state of the asset information window of the catalog.

Catalog.CategoryInformationConfiguration:
This property returns or sets the "Show All Fields" state of the category information window of the catalog.

See also InformationConfiguration

ShowAllObjectsInOneWindow As Boolean

This property returns or sets the "Show All Objects In One Window" value of the information window. "Show All Objects In One Window" controls wether Cumulus opens a new Information window for each asset / category.

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
This property returns or sets the "Show All Objects In One Window" state of the asset information window of the catalog.

Catalog.CategoryInformationConfiguration:
This property returns or sets the "Show All Objects In One Window" state of the category information window of the catalog.

See also InformationConfiguration

SynchronizeToSelection As Boolean

This property returns or sets the "SynchronizeToSelection" value of the information window. "SynchronizeToSelection" controls wether records / categories selected in the collection window will be displayed accordingly in the information window.

The behavior of this method depends on the origin of the InformationConfiguration object.

Catalog.AssetInformationConfiguration:
This property returns or sets the "Synchronize To Selection " state of the asset information window of the catalog.

Catalog.CategoryInformationConfiguration:
This property returns or sets the "SynchronizeToSelection " state of the category information window of the catalog.

See also InformationConfiguration

ViewEntry Object

The meaning of this object depends on the origin of the ViewEntry object.

CatalogCollection.DetailsViewConfiguration.Item:
The view entry object describes the properties of a column in the collection's details view.

CatalogCollection.ThumbnailViewConfiguration:
The view entry object describes the properties of a single field in the collection's thumbnail view.

- Application
- Bold
- CharSet
- Color
- ColumnWidth
- FieldUID
- FontName
- FontSize
- Italic
- Name
- Parent
- StrikeOut
- Underline

See also Overview

Group Object

The group object represents a group of users. This object allows the handling of the individual Groups elements.

Application
ID
Name
Parent

See also Overview

Groups Object

The Groups object can be used to enumerate all group objects. The Groups collection can be requested with Catalog.Groups.

Add
Application
Count
Item
ItemByID
Parent
Remove
RemoveByID

See also Overview

AssetStorage Object

The asset storage object is responsible for the asset access.

Active
Application
AssetStorageUID
Name
Parent

See also Overview

AssetStorages Object

The AssetStorages object can be used to enumerate all asset storage objects. An asset storage is responsible for the asset access. The AssetStorages collection can be requested with CatalogingOptions.AssetStorages.

- Application
- Count
- Item
- ItemByID
- Parent

See also Overview

CategoryTypeConstants

These constants describe the type of a category.

Possible values are:

NormalCategory	= 0
RelatedCategory	= 1 (read-only)
DirectoryCategoryMac	= 8
DirectoryCategoryWin	= 24
DirectoryCategoryURL	= 40

CentralAssetLocationUsageConstants

These constants describe the mode of the central asset location folder’s usage.

Possible values are:

Always	= 0
AsCatalogingOption	= 1
Never	= 2

SortDirectionConstants

These constants describe the catalog’s sort direction.

Possible values are:

Ascending	= 0
Descending	= 1

ViewConstants

These constants describe the collection’s viewing mode.

Possible values are:

Details	= 0
SmallThumbnails	= 1
MediumThumbnails	= 2
LargeThumbnails	= 3
CustomSizeThumbnails	= 4
PaletteView	= 5

ThumbnailQualityConstants

These constants describe the thumbnail quality for creating thumbnails.

Possible values are:

NormalQuality	= 0
HighQuality	= 1

ThumbnailSourceConstants

These constants describe the desired source for the creation of thumbnails.

Possible values are:

FromEntireAsset	= 0
FromStoredThumbnail	= 1
FromStoredThumbnailIfLargeEnough	= 2

CatalogingActionConstants

These constants describe the action that is used for the cataloging process.

Possible values are:

AddOnly	= 0
AddOrUpdate	= 1
UpdateOnly	= 2

PreviewSizeConstants

These constants describe the default size for opening preview windows.

Possible values are:

OriginalPreviewSize	= 0
MaximumPreviewSize	= 1
ManualPreviewSize	= 2

UnitConstants

These constants describe the display unit.

Possible values are:

uInch	= 0
uCentimeter	= 1
uMillimeter	= 2
uPoint	= 3
uPixel	= 4
uPica	= 5

OpenOnDoubleClickConstants

These constants describe what sort of window should be opened on double-clicking a record.

Possible values are:

PropertyWindow	= 0
PreviewWindow	= 1

LanguageConstants

These constants describe which language will be used for a special method.

Possible values are:

LanguageDefault	= 0
LanguageUSEnglish	= 1
LanguageGerman	= 2
LanguageJapanese	= 3
LanguageFrench	= 4
LanguageItalian	= 5
LanguagePortugueseBrazilian	= 6
LanguageSpanish	= 7
LanguageDutch	= 8
LanguageKorean	= 9

ListOfDataConstants

These constants describe what kind of information is returned on calling FieldDefinitions.ListOfData or Fields.ListOfData.

Possible values are:

ListOfFieldType	= 1
ListOfFieldUID	= 2
ListOfFieldName	= 3
ListOfShortID	= 4
ListOfHasValue	= 5
ListOfValue	= 6
ListOfAllowEdit	= 7

FindTokenConstants

These constants are used as arguments for the FindToken property.

Possible values are:

FindContains	= 0
FindStartsWith	= 1
FindIs	= 2
FindIsNot	= 3
FindDoesntContain	= 4
FindUnder	= 5
FindUpTo	= 6
FindFrom	= 7
FindOver	= 8
FindPrior	= 9
FindAfter	= 10
FindThumbSimilar	= 11
FindThumbNotSimilar	= 12
FindHasNoValue	= 13
FindHasValue	= 14
FindBoolTrue	= 15
FindBoolFalse	= 16
FindAnd	= 17
FindOr	= 18
FindCategoriesField	= 19

LayoutFieldTypeConstants

These constants describe the type of a catalog’s field.

Possible values are:

FieldTypeString	= 0
FieldTypeBool	= 1
FieldTypeInteger	= 2
FieldTypeDouble	= 3
FieldTypeDate	= 4
FieldTypeBinary	= 5
FieldTypePicture	= 6
FieldTypeEnum	= 7
FieldTypeAudio	= 8

AppPrefChangedEventConstants

These constants are used as arguments in the Application.GenericEvent handler in case of EventID = EV_UI_APP_PREF_CHANGED.

Possible values are:

AppPrefUnit	= 0
AppPrefLargeThumbnailSize	= 1
AppPrefMediumThumbnailSize	= 2
AppPrefSmallThumbnailSize	= 3
AppPrefLastRecentApplicatonList	= 4

WriteMetadataConstants

These constants are used as arguments in the Catalog.WriteMetadataToAssets property.

Possible values are:

WriteMetadataNever	= 0
WriteMetadataSilent	= 1
WriteMetadataWithUI	= 2

IndexKindTypeConstants

These constants describe the type of the record field index type.

Possible values are:

IndexKindNotIndexed	= 0
IndexKindSortIndexed	= 1
IndexKindContentIndexed	= 2

These values may be combined by adding, for example

IndexValue = IndexKindSortIndexed+ IndexKindContentIndexed

FindTypeConstants

These constants describe the type of the record search.

Possible values are:

FindNew	= 0
FindAgain	= 1
FindExtend	= 2

CharSetConstants

These constants describe the different language character sets of a font.

Possible values are:

ANSI_CHARSET	= 0
DEFAULT_CHARSET	= 1
SYMBOL_CHARSET	= 2
SHIFTJIS_CHARSET	= 128
HANGEUL_CHARSET	= 129
HANGUL_CHARSET	= 129
GB2312_CHARSET	= 134
CHINESEBIG5_CHARSET	= 136
OEM_CHARSET	= 255
JOHAB_CHARSET	= 130
HEBREW_CHARSET	= 177
ARABIC_CHARSET	= 178
GREEK_CHARSET	= 161
TURKISH_CHARSET	= 162
VIETNAMESE_CHARSET	= 163
THAI_CHARSET	= 222
EASTEUROPE_CHARSET	= 238
RUSSIAN_CHARSET	= 204
MAC_CHARSET	= 77
BALTIC_CHARSET	= 186

PreviewLinkTypeConstants

These constants are used as arguments in the Records.ExportToHTML method.

Possible values are:

HtmlNoLink	= 0
HtmlPreviewLink	= 1
HtmlFileLink	= 2
// Flags:	
HtmlFlagIncludeAudioFieldLink	= 256

FieldAccessConstants

These constants allow you to access the fields of a catalog (records and categories). These constants are GUIDs and can be used as FieldUID arguments in several properties and methods.

Some (predefined) values for the record fields are:

UID_REC_RECORD_NAME
UID_REC_THUMBNAIL
UID_REC_NOTES
UID_REC_COLOR_MODE
UID_REC_ASSET_TAG_CANNOT_BE_DELETED
UID_REC_ASSET_TAG_CANNOT_BE_MODIFIED
UID_REC_MAC_FILE_TYPE
UID_REC_MAC_FILE_CREATOR
UID_REC_HEIGHT
UID_REC_WIDTH
UID_REC_USER
UID_REC_ASSET_DATA_SIZE
UID_REC_ASSET_RSRC_SIZE
UID_REC_FORMAT
UID_REC_HORIZONTAL_RESOLUTION
UID_REC_VERTICAL_RESOLUTION
UID_REC_CREATING_SOFTWARE
UID_REC_ASSET_CREATION_DATE
UID_REC_ASSET_MODIFICATION_DATE
UID_REC_ASSET_TAG_CREATION_DATE
UID_REC_ASSET_TAG_MODIFICATION_DATE
UID_REC_FORMAT_SPECIFIC
UID_REC_STATUS
UID_REC_CATEGORIES (this is a **virtual** field)
UID_REC_WINDOWS_FILE_SERVER_NAME
UID_REC_WINDOWS_VOLUME_NAME
UID_REC_WINDOWS_FOLDER_NAME
UID_REC_WINDOWS_FILE_NAME
UID_REC_MAC_FILE_SERVER_NAME
UID_REC_MAC_FILE_SERVER_ZONE_NAME
UID_REC_MAC_VOLUME_NAME
UID_REC_MAC_FOLDER_NAME
UID_REC_MAC_FILE_NAME
UID_REC_URL_PROTOCOL_NAME
UID_REC_URL_HOST_NAME
UID_REC_URL_FOLDER_NAME
UID_REC_URL_FILE_NAME
UID_REC_ASSET_REFERENCE
UID_REC_THUMBNAIL_MEAN_VALUE
UID_REC_THUMBNAIL_STDDEV_VALUE

Some (predefined) values for the category fields are:

UID_CAT_NAME
UID_CAT_WIN_FOLDER_PATH
UID_CAT_URL_FOLDER_PATH
UID_CAT_MAC_VOLUME_ID

UID_CAT_MAC_DIRECTORY_ID
UID_CAT_TYPE
UID_CAT_ORIGINAL_ID
UID_CAT_PARENT_ID

Some (predefined) values for the Modules (see LinkEntry) are

UID_MOD_APPLICATION
UID_MOD_WINDOWS_FILE_SYSTEM
UID_MOD_CUMULUS_4_COMPATIBILITY_FILTER

Some (predefined) values for the asset storages (see AssetStorageUID) are

UID_AS_WIN_FILE
UID_AS_MAC_FILE
UID_AS_URL
UID_AS_VAULT
UID_AS_META_DATA
UID_AS_PIXEL_CONVERTER
UID_AS_DCS
UID_AS_OPI

Some (predefined) values for the database selectors (see OpenCatalog) are

UID_SELECTOR_CUE (Cumulus Engine)
UID_SELECTOR_ODBC (ODBC Connector)