Creating a Custom Metadata Synchronizer

An ESRI® White Paper • July 2002

ESRI ArcGIS™

ESRI 380 New York St., Redlands, CA 92373-8100, USA • TEL 909-793-2853 • FAX 909-793-5953 • E-MAIL info@esri.com • WEB www.esri.com

U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS

J-8651

Creating a Custom Metadata Synchronizer

An ESRI White Paper

Contents **Page**

Contents **Page**

Creating a Custom Metadata Synchronizer

Introduction

With the introduction of ArcCatalog™, ESRI® software users can now closely integrate metadata with their data. Metadata is a set of information that describes the characteristics of a data set and is frequently read by users who want to fully understand what a data set is and whether it will be of use to them.

Users can have ArcCatalog generate metadata automatically. In this process, ArcCatalog reads a data set to harvest its properties and writes these to the metadata. This process is called synchronization.

In ArcCatalog at ArcInfo™ 8.0.1 and 8.0.2, the metadata created by synchronization was formatted according to the American metadata standard, the Content Standard for Digital Geospatial Metadata (CSDGM), maintained by the Federal Geographic Data Committee (see http://www.fgdc.gov/metadata for details). At release 8.1, a new set of objects was added to ArcGIS™ that allows users to customize the metadata that is created during synchronization. This white paper describes these new objects and shows how they may be used.

What Is Synchronization?

Synchronization is the process by which properties of a data set are read from the data set and written into its metadata. The purpose of synchronization is twofold:

1. To complete as much of the metadata as possible automatically, thereby minimizing the amount of work that the user has to do to completely document a data set.

2. To make sure that the metadata is kept up-to-date with changes to the data set.

Within ArcCatalog, synchronization may be triggered by any of the following:

- When a user browses to a data set in ArcCatalog with the Metadata tab selected and that data set does not have metadata, a synchronize is triggered and metadata is created for the data set. (Note that the user can turn this automatic update option off using the Metadata tab found under Tools -> Options in ArcCatalog.)

- When a user browses to a data set in ArcCatalog with the Metadata tab selected and that data set has metadata, a synchronize is triggered and metadata is updated for the data set. (Note that the user can turn this automatic update option off using the Metadata tab found under Tools -> Options in ArcCatalog.)

- When a user clicks the Create/Update metadata button in ArcCatalog, a synchronize takes place and the metadata for the currently selected data set is created (if metadata does not exist) or updated (if it exists).

Using ArcObjects™, a programmer can also initiate synchronization using the *IMetadata::Synchronize* method. This is implemented by the set of *GxObject*s*;* the set of *Name* CoClasses; and the *Tin, GeometricNetwork, RasterData setName,* and *RasterBandName* coclasses.

How Synchronization Works

In ArcCatalog, special objects named metadata synchronizers are used to write metadata when synchronization occurs. ArcCatalog comes shipped with three metadata synchronizers: the *FGDCSynchronizer* CoClass, which writes metadata according to the FGDC standard; the *GNSynchronizer* CoClass, which writes metadata according to the ISO standard; and the *GNSynchronizer* CoClass, which is used to synchronize metadata for the Geography Network℠. A later section in this paper shows how you can create a custom metadata synchronizer so that you can write your own tags to the metadata.

Each metadata synchronizer must be registered using ArcGIS software's component category manager before it can be used to write metadata. The collection of registered metadata synchronizer objects is managed by a dedicated object called the synchronization manager. Figure 1 below describes how the synchronization manager works.

1. A synchronize is triggered, as described in the previous section.

2. A property of the selected data set is sent to the *Update* method of the synchronization manager.

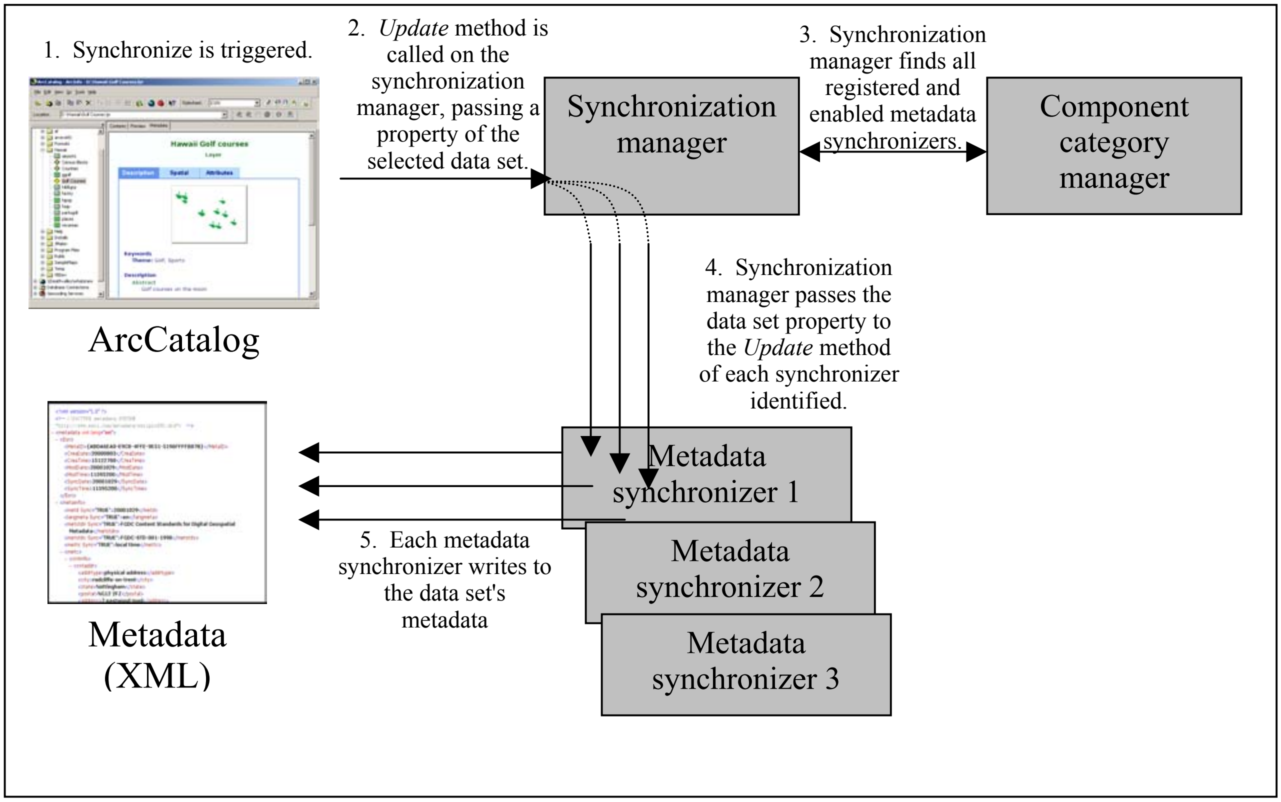
3. The synchronization manager must find out which metadata synchronizers to use to write the metadata. It does this by identifying what synchronizers have been registered using the component category manager. Each registered synchronizer is enabled or disabled by setting properties of the synchronizer through an interface on the synchronization manager, as described later. The synchronization manager queries each registered synchronizer, and the following steps are implemented for each synchronizer that is enabled.

4. The synchronization manager passes the property of the data set that it received in step 2 to each enabled metadata synchronizer.

5. The *Update* method of each metadata synchronizer contains code that processes the incoming object through the synchronization manager and defines metadata from this information. The *Update* method is discussed in more detail in a following section.

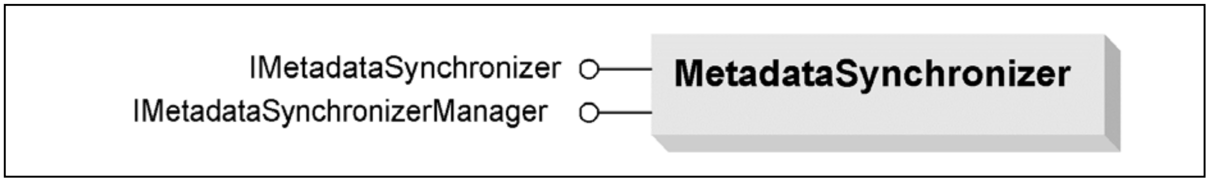
6. Steps 2 to 5 are repeated with other properties of the data set until all properties have been processed.

Figure 1
Mechanics of the Synchronization Process


1. Synchronize is triggered.
2. *Update* method is called on the synchronization manager, passing a property of the selected data set.
Synchronization manager
3. Synchronization manager finds all registered and enabled metadata synchronizers.
Component category manager
ArcCatalog
4. Synchronization manager passes the data set property to the *Update* method of each synchronizer identified.
Metadata synchronizer 1
Metadata synchronizer 2
Metadata synchronizer 3
5. Each metadata synchronizer writes to the data set's metadata
Metadata (XML)

The MetadataSynchronizer CoClass

This section looks in depth at the MetadataSynchronizer CoClass. The MetadataSynchronizer CoClass is part of ArcObjects, the set of Component Object Model (COM) objects that are used to develop applications that work with ArcGIS. The CoClass has two interfaces, which are used by programmers to create and manage metadata synchronizers. These interfaces are shown in the object model diagram below.



1. *IMetadataSynchronizer* defines the properties of a metadata synchronizer and is used to develop a custom synchronizer.

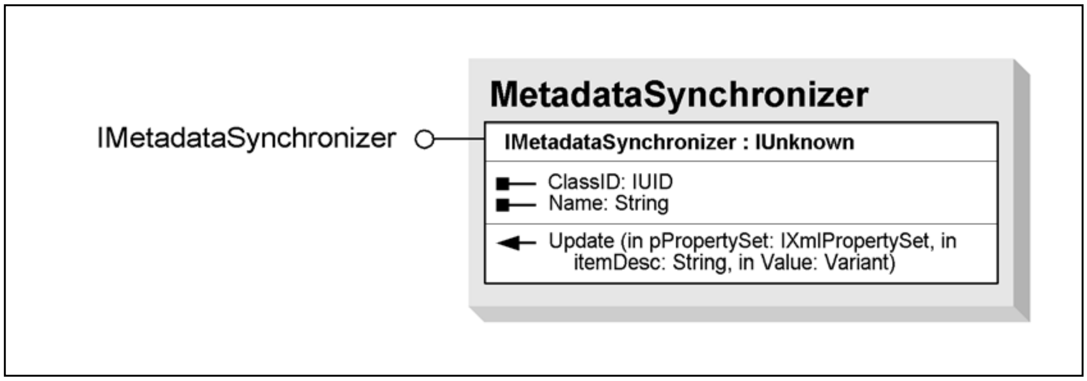
2. *IMetadataSynchronizerManager* is used by the synchronization manager to manage the metadata synchronizer objects that are registered on a user's system.

The IMetadataSynchronizer Interface

Interface *IMetadataSynchronizer* defines the characteristics and behavior of a metadata synchronizer. It has two properties, *ClassID* and *Name;* and one method, *Update*. These are used as follows:

- **ClassID—**A UID object, which must be unique to the synchronizer.
- **Name—**A string that identifies the synchronizer. This name is used when managing metadata synchronizers through the *IMetadataSynchronizerManager* interface.
- **Update—**This method is used to define the structure of the metadata that is written during synchronization. It is called once for each property of the data set during synchronization. Each time, the following three objects are passed to the method:

  1. **pPropertySet—**An XML property set representing the metadata of the data set that is being synchronized.
  2. **itemDesc—**A string that describes what object is being passed in as the third argument. (See the table below.)
  3. **Value—**An object that is a parameter of the data set whose metadata is being synchronized.


MetadataSynchronizer
IMetadataSynchronizer
IMetadataSynchronizer : IUnknown
ClassID: IUID
Name: String
Update (in pPropertySet: IXmlPropertySet, in itemDesc: String, in Value: Variant)

Code in the *Update* method queries the Value object passed in by the synchronization manager to retrieve whatever properties need to be written to the metadata. These are then written into the metadata using ArcObjects methods on the XmlPropertySet.

Inputs to the MetadataSynchronizer *Update* Method

Table 1 summarizes what objects are passed into the *Update* method by the synchronization manager. The *Value* column lists the type of the object passed in. The *itemDesc* column lists the string identifier passed in. When working with the *Update* method, the string in *itemDesc* is used to identify what the object in *Value* is.

Table 1
Arguments Passed In to IMetadataSynchronizer::Update

itemDesc	Value	Description
Boilerplate	Nothing	A special case that is called only the first time synchronization occurs. This may be used to add boilerplate text, such as documentation hints or fixed contact information for your organization, that should not be changed again by synchronization.
CoverageEntity[i]	IArcInfoTable	Provides access to INFO™ tables and a coverage feature class's feature attribute table. Used to record attribute information and to count the number of records or features.
CoverageFeatureClass[i]	ICoverageFeatureClass	Provides information about a coverage feature class including the type of feature class and whether it has an attribute table or topology. Used to record feature information.
DatasetLocation	String	Location of data set on disk, or for ArcSDE™ data sets, connection information for accessing the geodatabase.
DatasetName	String	Name of the data set derived from either the file name or the table name.
DatasetSize	String	Size of the data set on disk. (Not used for objects stored in a geodatabase.)
DDExtent	IEnvelope	Envelope containing the data set's geographic data. Used to record its extent in decimal degrees.
Entity[i]	IClass	Provides access to an object class such as a table or feature class. Used to record full attribute information.
EntityBrief[i]	IClass	Provides access to an object class such as a table or feature class. Used to record brief entity information for the feature classes contained in a feature data set.
Environment	String	Operating system, software name, and version of the computer.
FeatureClass[i]	IFeatureClass	Provides access to a feature class. Used to record feature information such as feature and geometry type.
GeoForm	String. One of: "raster digital data", "remote sensing image", "tabular digital data", "vector digital data"	Mode in which the spatial data is represented.
GeometryType	String One of: "Vector", "Raster"	Type of geometry stored in the data set.
Language	String	Language of the data and the metadata. Derived from the operating system's default input locale.
MetadataDate	String	The current date. Used to record when the metadata was last updated.

Table 1 (continued)
Arguments Passed In to IMetadataSynchronizer::Update

itemDesc	Value	Description
MetadataStandard	String	The name of the metadata standard supported by ArcCatalog: version 2 of the CSDGM. Used to record information about the standard to which the metadata was created.
NativeExtent	IEnvelope	Envelope containing the data set's geographic data. Used to record its actual extent, either in projected or decimal degree coordinates.
NativeForm	String	Type of the data set, for example, "Shapefile", "Personal Geodatabase Table", or "Raster Dataset". This string is the same as that displayed in the Type column in ArcCatalog contents view.
NetworkRule[i]	IRule	Provides information about the connectivity rules in a geometric network.
NetworkSchema	INetSchema	Provides information about the schema of a geometric network such as element classes, ancillary roles, and weights.
OperatingSystem	String	The name of the operating system on the computer used to create or update the metadata (duplicated in the itemDesc *Environment*).
RasterBand	IRasterBand	Provides access to information about a raster band including its attribute table and color map.
RasterDataset	IRasterDataset (may also support IRasterBandCollection)	Provides information about a raster data set such as its format and compression type.
Relationship[i]	IRelationshipClass	Used to record detailed information about a relationship. Information such as the relationship's origin, destination, and cardinality is included.
RelationshipBrief[i]	IRelationshipClass	Used to record brief relationship information for objects that participate in a relationship.
Software	String	The name and version of the software used to create or update the metadata (duplicated in the itemDesc *Environment*).
SpatialReference	ISpatialReference	Provides access to the data set's spatial reference.
Tin	ITin	Provides access to information about a TIN data set.

Note that for item descriptions that are shown in the table with an index number such as itemDesc[i], several different objects of this type may be sent to the metadata synchronizer. For example, a coverage has several feature classes. To record the coverage's attributes, *Update* will be called once for each feature class that has a feature attribute table. Each time, the itemDesc parameter will be CoverageEntity[i] and the Value parameter will be a different *IArcInfoTable* object. If the coverage has one polygon and two region feature classes, *Update* will be called with the itemDesc values CoverageEntity[0], CoverageEntity[1], and CoverageEntity[2] with the appropriate *IArcInfoTable* objects.

The objects that are passed in will vary from one type of data set to another. Not all of these objects will be passed in for all data sets; the list of objects that is passed depends on the type of the data set (see the description column in the table for guidance). For example, CoverageEntity objects apply only to ArcInfo coverages and INFO tables. If the data set is a dBASE table, the following list of item descriptions will be passed: Boilerplate, DatasetLocation, Name, DatasetSize, Entity[i], Environment, GeoForm, Language, MetadataDate, MetadataStandard, NativeForm, OperatingSystem, and Software. For a shapefile, the DDExtent, FeatureClass[i], GeometryType, NativeExtent, and SpatialReference objects would also be passed.

There is a useful ESRI white paper called *Synchronization in ArcCatalog* that describes in detail how ArcCatalog populates FGDC metadata during synchronization. This white paper is available on the Internet from ArcOnline (http://arconline.esri.com).

The IMetadataSynchronizerManager Interface

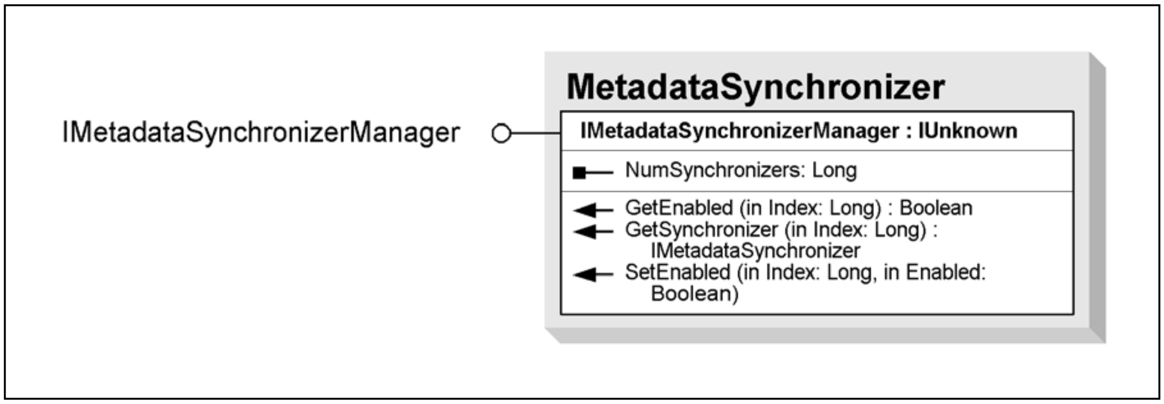
This interface is used when working with the synchronization manager. It allows you to enable or disable any of the registered metadata synchronizers and thus specify which synchronizer will write metadata during synchronization. The following property and methods are supported.

- **NumSynchronizers—**A long integer that gives the number of metadata synchronizer objects that are registered on the machine. The objects are registered in the *ESRI Metadata Synchronizers* component category.

- **GetEnabled—**A Boolean that is True if the synchronizer referenced by the index passed in is currently enabled.

- **GetSynchronizer—**The metadata synchronizer object referenced by the index passed in. It is used to get a reference to the synchronizer itself.

- **SetEnabled—**Sets the enabled/disabled state of the synchronizer referenced by the index passed in to the first argument. The second (Boolean) argument sets the state; if set to True then the synchronizer is enabled, if False then it is disabled.


IMetadataSynchronizerManager
MetadataSynchronizer
IMetadataSynchronizerManager : IUnknown
NumSynchronizers: Long
GetEnabled (in Index: Long) : Boolean
GetSynchronizer (in Index: Long) : IMetadataSynchronizer
SetEnabled (in Index: Long, in Enabled: Boolean)

Working with the Synchronization Manager

This section illustrates how you can work with the metadata synchronization manager using code to disable a specific metadata synchronizer. The *MetadataSynchronizer* object is a singleton, meaning that only one instance of this object exists. Thus, you can declare a reference to the *MetadataSynchronizer* directly in code at any time, and you will always get a reference to the same object.

Once you have a reference to the *MetadataSynchronizer,* you can query the interface of this object for the *IMetadataSynchronizerManager* interface and then use the *SetEnabled* method to enable or disable any of the metadata synchronizers. *SetEnabled* requires you to pass the index number of the synchronizer you wish to enable. Since there is no way for you to tie the index number to a specific synchronizer, you must loop through all of the synchronizers and test the *Name* of each; the name of the metadata synchronizer is defined when it is created.

The following Visual Basic® for Applications (VBA) code example will disable the FGDC Synchronizer, the metadata synchronizer that is shipped with ArcCatalog. The name of this synchronizer is "FGDC CSDGM." To use the code, open the Visual Basic editor in ArcCatalog or ArcMap™, paste the code in the box into a VBA module, and run it.

```
Public Sub DisableFGDCSync()

   ' Reference the MetadataSynchronizer singleton
   Dim MDSync As New esriCore.MetadataSynchronizer

   ' QI the Synchronizer for the Sync manager interface
   Dim pMDSyncM As esriCore.IMetadataSynchronizerManager
   Set pMDSyncM = MDSync

   ' Loop through all enabled synchronizers
   Dim i As Long
   For i = 0 To pMDSyncM.NumSynchronizers - 1

      ' Test the synchronizer at the current index position.
      ' If it is the FGDC synchronizer and is enabled, then disable it
      If  (pMDSyncM.GetSynchronizer(i).Name = "FGDC CSDGM") _
      And (pMDSyncM.GetEnabled(i)) Then
         MsgBox "Disabling the FGDC synchronizer..."
         pMDSyncM.SetEnabled i, False
      End If
   Next i
End Sub
```

How to Implement a Custom Synchronizer

This section discusses the recommended steps to follow when implementing a custom synchronizer.

1. Decide what information you wish to have written into the synchronized metadata. You should consider the following:

   a. Are you going to follow a published metadata standard or profile? Doing so will help users of the metadata to understand the elements that you are creating because they will be able to refer to documentation for an explanation of the metadata content.

   b. Where are you going to get the metadata required? The easiest way to do this is to take each of the metadata elements you wish to create and decide which of the objects passed to the synchronizer's *Update* method (listed in Table 1) have properties that can be queried to retrieve the information required.

   c. Many metadata standards require the user to collect documentation about the data set, which cannot be retrieved from the properties that are available in the *Update* method (for example, an abstract that describes the data set). If you wish to document these elements in the metadata, for example to add a documentation hint that indicates which elements are mandatory in the standard you are using, do this in the code that runs when passed the *Boilerplate* userDesc in the *Update* method.

      If the synchronizer will only be used within a specific organization, then metadata content may be written in the *Boilerplate*. For example, contact details that are not changed at any other time, such as an organization's name and address, can be written once for all new metadata so that it does not have to be entered manually.

      The *Boilerplate* is suitable for these metadata elements because it is only called once, when the metadata is first synchronized. Therefore, if the user edits a value it is not in danger of being overwritten during subsequent synchronizations.

2. Write an ActiveX® DLL that implements the *IMetadataSynchronizer* interface. This may be written in any COM-compliant language such as C++, Visual Basic, or Delphi™. The following section gives a code example written in Visual Basic.

3. Register the DLL on the machine on which the synchronizer is to be used. From the command line, you can use `regsvr32 <DLLname>` to do this.

4. Run the ArcGIS Component Category Manager (categories.exe, located in the folder arcexe81/bin). In the Component Category Manager dialog, do the following:

   a. Scroll down and highlight ESRI Metadata Synchronizers.

   b. Click Add Object.
   c. In the file browser, navigate to the DLL and click Open.

d. In the Add Objects dialog, you will see the exposed classes in the DLL listed (often there will just be one). Make sure that the class containing your implementation of *IMetadataSynchronizer* is checked. Click OK to close this dialog.

e. Back in the Component Category Manager, if you expand the node ESRI Metadata Synchronizers, you should see your class listed. Click OK to close the dialog. (You may also see the class `esriCore.FGDCSynchronizer` listed. This is the default synchronizer for ArcCatalog, which contains the synchronization information in support of the FGDC metadata standard.)

5. The next time you start ArcCatalog and resynchronize metadata for a data set, you can inspect the metadata to see that the metadata defined by your synchronizer is present. You can do this by looking at the metadata with the XML stylesheet in ArcCatalog.

Visual Basic Code Example

The following code example is written in Visual Basic 6 to illustrate some of the concepts discussed above. To use this code, open Visual Basic and create a new project of type *ActiveX DLL*. Next, open the project references window, select ESRI Object Library, and click OK. The new project will have an empty Class Module into which you may paste the following code.

Note: Before compiling the DLL, you must set the names of the project and the class module in the code for property *ClassID* and give a name for the synchronizer in the code for property *Name*. See the comments in the code below for explanations.

```
Implements esricore.IMetadataSynchronizer

Private Property Get IMetadataSynchronizer_ClassID() As esriCore.IUID

    ' An ID is required to distinguish this synchronizer uniquely.
    ' Safest is to use the UID of this VB Class module
    Dim myUID As New esriCore.UID

    ' *** Enter project name and class name here ***
    Set myUID.Value = "ProjectName.ClassModuleName"
    IMetadataSynchronizer_ClassID = myUID
End Property

Private Property Get IMetadataSynchronizer_Name() As String

    ' A string describing the synchronizer, which is used to identify it
    ' when working with the synchronization manager

    ' *** Enter synchronizer name here ***
    IMetadataSynchronizer_Name = "My synchronizer"
End Property
```

```
Private Sub IMetadataSynchronizer_Update(ByVal pPropertySet As _
    esriCore.IXmlPropertySet, ByVal itemDesc As String, ByVal Value As Variant)

    ' Handles writing the metadata.  Tests itemDesc using a Case statement,
    ' then makes appropriate tests on the Value object and writes results to the
    ' propertySet.

    Dim lBracketPos As Long
    Dim sEltDesc As String
    Dim sNumPlus As String
    Dim lEltCount As Long

    ' Some item descriptions have bracketed counts, e.g. Entity[0].  Strip the
    ' count off into a variable
    lBracketPos = InStr(1, itemDesc, "[")
    If lBracketPos > 0 The
        sNumPlus = Right(itemDesc, Len(itemDesc) - lBracketPos)
        lEltCount = CLng(Left(sNumPlus, Len(sNumPlus) - 1))
    Else
        sEltDesc = itemDesc
    End If

    Select Case sEltDesc
    Case "Boilerplate"
        ' Write documentation hints for metadata documentation that is not
        ' described in properties of the data
        pPropertySet.SetPropertyX "abstract", _
            "REQUIRED: Summary description of the data set", esriXPTText, _
            esriXSPAAddOrReplace, False

    Case "DDExtent"
        ' Data set extent in decimal degrees (type IEnvelope)

        ' Must QI into an object of correct type as properties on the Value
        ' object will return Null
        Dim pBoundBox As esricore.IEnvelope
        Set pBoundBox = Value

        ' Check that the envelope is not empty so we don't write zero tags
        If Not pBoundBox.IsEmpty Then
            pPropertySet.SetPropertyX "boundrec/west", _
            CStr(pBoundBox.XMin), esriXPTText, esriXSPAAddOrReplace, True
            pPropertySet.SetPropertyX "boundrec/east ", _
            CStr(pBoundBox.XMax), esriXPTText, esriXSPAAddOrReplace, True
            pPropertySet.SetPropertyX "boundrec/north", _
            CStr(pBoundBox.YMax), esriXPTText, esriXSPAAddOrReplace, True
            pPropertySet.SetPropertyX "boundrec/south", _
            CStr(pBoundBox.YMin), esriXPTText, esriXSPAAddOrReplace, True

            ' Add property explaining what spatial ref system the coords are in
            pPropertySet.SetAttribute "boundrec", "SpatRefSystem", _
            "LatLong", esriXSPAAddOrReplace
        End If
```

```
   Case "FeatureClass"
       ' The Geometric feature classe(s) of the data set
       ' (type: IFeatureClass[i])  lEltCount gives the number of this feat class

       Dim pFeatClass As IFeatureClass
       Set pFeatClass = Value

       ' Test the shape type and write a description accordingly.
       ' Note that, in this example, not all alternatives are tested
       Select Case pFeatClass.ShapeType
       Case esriGeometryPoint
          strOutput = "Point"
       Case esriGeometryMultipoint
          strOutput = "Multipoint"
       Case esriGeometryLine
          strOutput = "Line"
       Case esriGeometryPolygon
          strOutput = "Polygon"
       Case Else
          strOutput = ""
       End Select

       ' Only write a result if we have something to write
       If Len(strOutput) > 0 Then

          ' Use the element count surrounded by braces to allow more than
          ' one featclass tag to be written to the metadata
          pPropertySet.SetPropertyX "featclass[" & CStr(lEltCount) & _
          "]/ShapeType", strOutput, esriXPTText, esriXSPAAddOrReplace, True
       End If

   End Select
End Sub
```

Other Considerations

A custom synchronizer allows you to add content to metadata. If you want others to be able to interact with and use this metadata, you may need to add more features to ArcGIS. This section summarizes some of these considerations. You can read more about these areas on ArcOnline (http://arconline.esri.com) and see code examples in the folder arcexe81\ArcObjects Developer Kit\Samples\Metadata, found under your ArcGIS installation directory. (Note that you must have performed a custom or complete install of ArcGIS to have the samples installed.)

Metadata Stylesheet

In ArcCatalog, a stylesheet is used to define what pieces of metadata are displayed to the user when they view metadata and how this information is displayed. When viewing metadata, the user may change the way that the metadata is displayed by selecting a stylesheet from the drop-down list on the metadata toolbar.

ArcCatalog is shipped with a number of stylesheets that display metadata stored in FGDC CSDGM format. For you to see the metadata you have created with a custom synchronizer, you will need to create a stylesheet that looks for and displays your metadata, or alternatively add commands into one of the FGDC stylesheets provided to display your metadata alongside the FGDC metadata. You do not need to create a stylesheet if you do not want users to see your metadata in ArcCatalog.

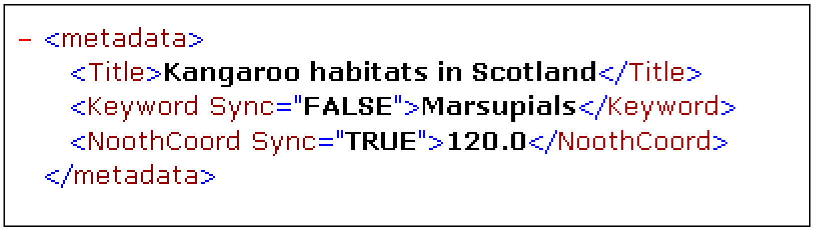
Custom Metadata Editor

A metadata editor gives users an interface that they can use to change the content of the metadata. The metadata editor shipped with ArcCatalog allows users to edit all metadata elements in the FGDC standard; therefore, they can use this to change any element that was written using the FGDC synchronizer. You should consider whether you want users to be able to edit metadata created with your synchronizer, and if so, then implement a custom metadata editor that allows them to do so.

It is recommended that an editor be provided where you have added documentation hints into the metadata in the *Boilerplate* section of the *Update* method, so that users can replace your hints with the proper metadata content.

If you do allow users to change metadata that you are synchronizing in your code, then you should consider writing your editor so that it prevents the synchronizer from overwriting an element that the user has changed. (The FGDC metadata editor in ArcCatalog is designed with the principle that if a user has made a conscious decision to edit a metadata element, then the user's decision overrides any automatic synchronization.)

ArcCatalog controls how a metadata element is updated during synchronization using an attribute called *Sync,* which is set on metadata elements to indicate whether the synchronizer should change them or not. If the attribute is present on an element, and it is set to True, then the element may be changed by the synchronizer. If the *Sync* attribute is not present or is False, then the element will not be changed. In the following XML example, the elements Title and Keyword will not be changed by the synchronizer, but the element NorthCoord could be.



```
- <metadata>
    <Title>Kangaroo habitats in Scotland</Title>
    <Keyword Sync="FALSE">Marsupials</Keyword>
    <NoothCoord Sync="TRUE">120.0</NoothCoord>
  </metadata>
```

In the Visual Basic code example given in the previous section, metadata was written using the method *IXmlPropertySet::SetPropertyX*. The last argument to this method is a Boolean that indicates whether the *Sync* flag is analyzed when setting the element in question—in the example, this is always set to True. It is recommended that this parameter is always set to True in your synchronizer code (except when setting documentation hints in the *Boilerplate* code, as described earlier), which means that the *Sync* attribute will be tested before the metadata element is changed by the synchronizer. If the element does not exist, then it will be created with a *Sync* attribute set to True. (Note: If you set the last argument in *SetPropertyX* to False, then the *Sync* attribute is ignored and the metadata is changed regardless of the value of the *Sync* attribute.)

Given the above, it is therefore recommended that your custom editor code sets the *Sync* value to False if the user has changed an element that is synchronized by your synchronizer. If you do not do this, then the synchronizer will reset the value typed by the user on the next synchronization. To set the attribute to False, you may use the method *IXmlPropertySet::SetAttribute*. For example, the following line of VB code will set the *Sync* attribute to False for the NorthCoord element in the example above.

```
pXmlPSet.SetAttribute "NorthCoord", "Sync", "FALSE", esriXSPAReplaceIfExists
```

Custom Metadata Import and Export

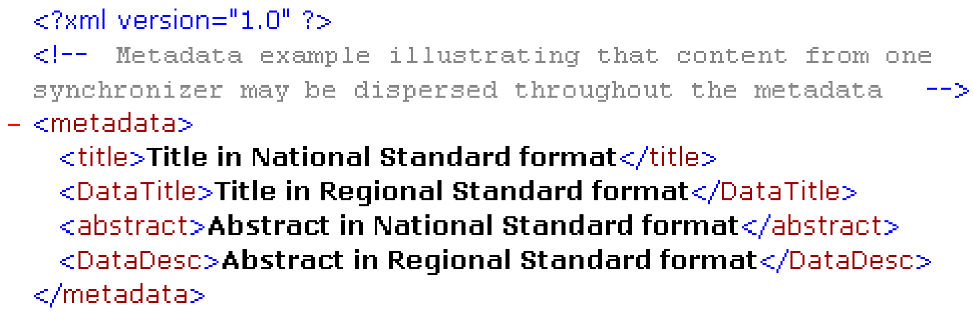
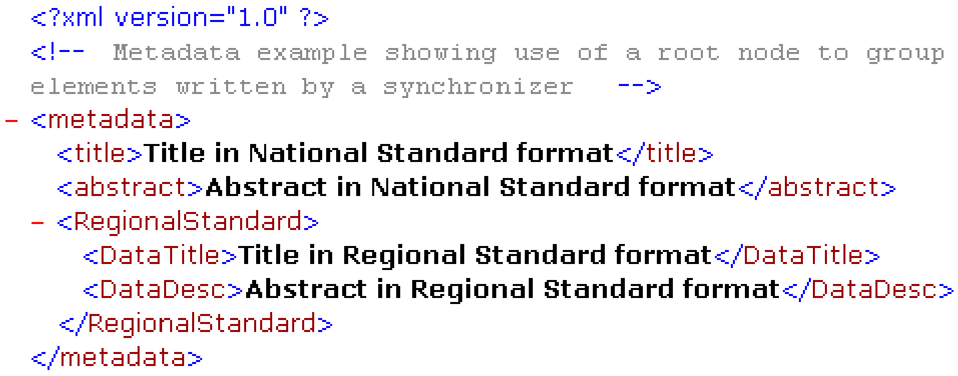
ArcCatalog has support for users to import and export metadata to other formats so that it may be shared with users running other metadata systems. ArcCatalog is shipped with an importer that reads XML that has any content, an exporter that writes HTML and XML with any content, plus a set of importers and exporters that support FGDC format metadata only. If you wish to provide utilities for users to import and export metadata according to the standard that is implemented in your synchronizer, then you will need to write a custom import and/or export tool in support of this.

Limitations

The following limitations should be kept in mind when working with custom metadata synchronizers.

Metadata from One Synchronizer May Be Dispersed in the XML

If you have implemented a number of synchronizers, then each will write metadata into the same XML file. It is not possible to have separate metadata files for a data set. Within the XML file, it is not possible to specify what ordering is used when elements are written to the metadata; therefore, you cannot be sure that the metadata elements created from one synchronizer will always be placed in adjacent positions in the metadata file. If you have written a stylesheet for users to display your metadata and a custom exporter to write it out then this will not be a problem. The stylesheet and exporter can define the ordering of elements that the user sees, and the user does not see the source metadata file in these cases. (If you plan to validate the XML using a DTD, then you will need to ensure that the ordering of elements in the XML corresponds with the ordering defined by the DTD.) When writing the metadata, if you wish to group the elements from one synchronizer together, then the simplest way to support this is to always use a common "root" element to group your metadata separately from any other metadata that may be written into the file. (An XML file cannot have more than one root node, so the elements you create to group your metadata must always be under the <metadata> root node.) The simple XML examples in the diagram below illustrate this idea.

XML file illustrating how metadata from two synchronizers may be dispersed	 <?xml version="1.0" ?> <!-- Metadata example illustrating that content from one synchronizer may be dispersed throughout the metadata --> - <metadata> <title>**Title in National Standard format**</title> <DataTitle>**Title in Regional Standard format**</DataTitle> <abstract>**Abstract in National Standard format**</abstract> <DataDesc>**Abstract in Regional Standard format**</DataDesc> </metadata>
XML file illustrating how metadata from two synchronizers may be distinguished using elements under the root node	 <?xml version="1.0" ?> <!-- Metadata example showing use of a root node to group elements written by a synchronizer --> - <metadata> <title>**Title in National Standard format**</title> <abstract>**Abstract in National Standard format**</abstract> - <RegionalStandard> <DataTitle>**Title in Regional Standard format**</DataTitle> <DataDesc>**Abstract in Regional Standard format**</DataDesc> </RegionalStandard> </metadata>

XML Headers Cannot Be Written

It is not possible to write to the header of the metadata using methods on the *XmlPropertySet* CoClass. Consequently, you will not be able to write any sort of documentation into the header of the XML file. For this reason, it is suggested that you do not use schemas and namespaces to distinguish the metadata tags in your custom synchronizer from those written in another synchronizer because you will not be able to write the *xmlns* attribute into the <metadata> root element tag through ArcObjects. It is invalid to have an XML file that uses schemas and namespaces but does not have a corresponding *xmlns* attribute. Doing this would be creating metadata that cannot be displayed in ArcCatalog.

Some FGDC Metadata Elements Will Always Be Present

It is possible to disable the FGDC metadata synchronizer that is shipped as the default synchronizer with ArcCatalog. Even if you do this, however, some FGDC format metadata will still be written to the output during synchronization (what is written depends on the data format). This limitation will be removed in later releases of ArcCatalog but is present at release 8.1.

If you write custom stylesheets and exporters for the metadata supported in your synchronizer, then users need not be aware of this fact because the FGDC metadata elements will not be seen by them.

For more than 30 years ESRI has been helping people manage and analyze geographic information. ESRI offers a framework for implementing GIS technology in any organization with a seamless link from personal GIS on the desktop to enterprisewide GIS client/server and data management systems. ESRI GIS solutions are flexible and can be customized to meet the needs of our users. ESRI is a full-service GIS company, ready to help you begin, grow, and build success with GIS.

Corporate

ESRI
380 New York Street
Redlands, California
92373-8100, USA
Telephone: 909-793-2853
Fax: 909-793-5953

For more information
on ESRI, call

1-800-447-9778

(1-800-GIS-XPRT)

or contact an ESRI reseller near you.

Send e-mail inquiries to

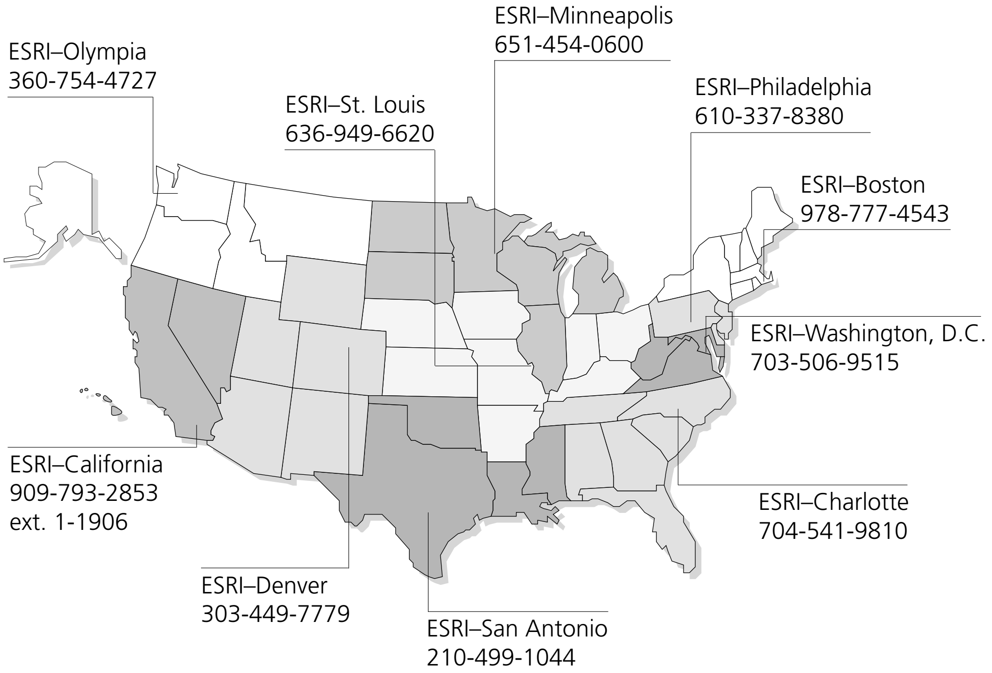
info@esri.com

Visit ESRI's Web page at

www.esri.com

Outside the United States,
contact your local ESRI distributor.
For the number of your distributor,
call ESRI at 909-793-2853,
ext. 1235,
or visit our Web site at
www.esri.com/international

Regional Offices



ESRI–Olympia
360-754-4727

ESRI–St. Louis
636-949-6620

ESRI–Minneapolis
651-454-0600

ESRI–Philadelphia
610-337-8380

ESRI–Boston
978-777-4543

ESRI–Washington, D.C.
703-506-9515

ESRI–California
909-793-2853
ext. 1-1906

ESRI–Denver
303-449-7779

ESRI–San Antonio
210-499-1044

ESRI–Charlotte
704-541-9810

International Offices

Australia	613-9867-0447
Belgium/Luxembourg	32-2-460-7480
Bulgaria	359-2-964-0850
Canada	416-441-6035
France	33-1-46-23-6060
Germany/Switzerland	49-8166-677-0 41-1-360-2460
Hong Kong	852-2730-6883
Hungary	361-428-8040
India	91-11-620-3802
Indonesia/Malaysia	62-21-570-7685 603-7874-9930
Italy	39-06-406-96-1
Japan	81-3-3794-6681
Korea	82-2-571-3161
Netherlands	31-10-217-0700
Poland	48-22-825-9836
Portugal	351-2-1-781-6640
Romania	40-1-231-13-81
Singapore	65-742-8622
Spain	34-91-559-4375
Sweden	46-23-755-400
Thailand	66-2-678-0707
United Kingdom	44-1296-745-500
Venezuela	58-212-285-9394

Place ESRI business partner or distributor address here.

No. GS-35F-5086H

Printed in USA