Introducing Bentley Map VBA Development

Jeff Bielefeld

Bentley®
Sustaining Infrastructure

Session Overview

- **Introducing Bentley Map VBA Development** - In this session attendees will be provided an introductory look at what is required to begin developing custom MicroStation VBA applications for the Bentley Map platform. An overview of the “XFM Feature Toolkit” (XFT) will be provided to familiarize each attendee with the Bentley Map XFM object model. A number of sample Bentley Map VBA applications will be discussed and demonstrated, showing attendees common techniques used to create, locate and edit XFM feature instances.

Agenda

- Answer some of the most “Frequently Asked Questions” related to the Bentley Map product line.
- XML-Based Feature Modeling (XFM) Overview
- XFM Feature Toolkit (XFT) Overview
- Bentley Map VBA Examples

Bentley Map

Frequently Asked Questions

Frequently Asked Questions

- **Question** - Why is the Bentley Map data model different than that of MicroStation or the one used for years in the MicroStation GeoGraphics product?
- **Answer** - While MicroStation has always provided an industry leading CAD engine, users requiring geospatial functionalities often need additional data modeling and geo-processing capabilities. For some 15 years, MicroStation GeoGraphics filled that gap by extending MicroStation into a “feature-based” system. Several years ago, users began requesting more extensible data modeling capabilities, thus the “**X**ML-Based **F**eature **M**odeling (**XFM**)” framework was developed.

Frequently Asked Questions

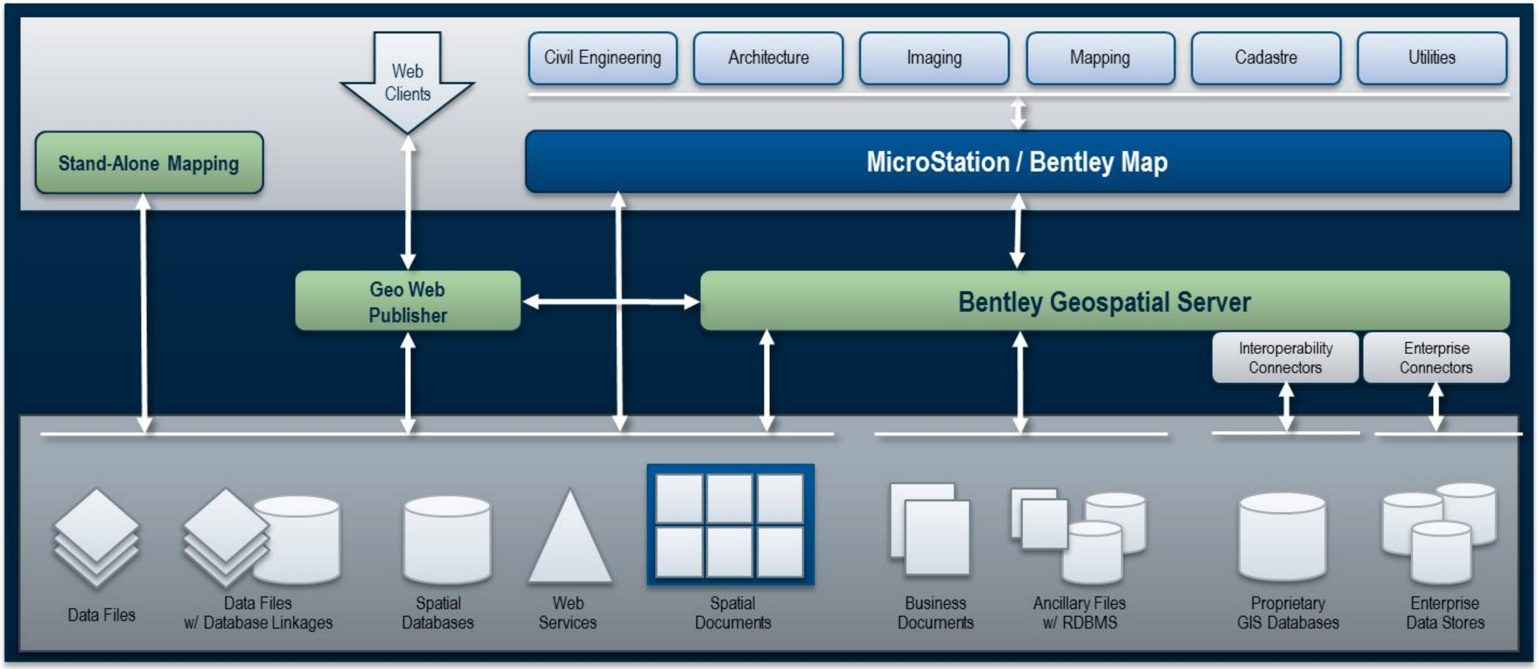
- **Question** – What is XFM?
- **Answer** – As previously mentioned, the term XFM stands for "**X**ML-Based **F**eature **M**odeling" which provides an extensible XML-based metadata driven framework upon which geospatial feature classes, their business properties and behaviors can be modeled. Let's take a brief look at some of the key benefits of the XFM data modeling platform.

XFM Key Benefits

- Provides an extensible framework upon which next generation infrastructure applications can be developed.
- Provides hierarchical data modeling capabilities.
- Provides support for disconnected workflows.
- Provides ability to model data from emerging data standards.
- Improves data interoperability capabilities.
- Integrated MicroStation undo/redo support.

XFM Key Benefits

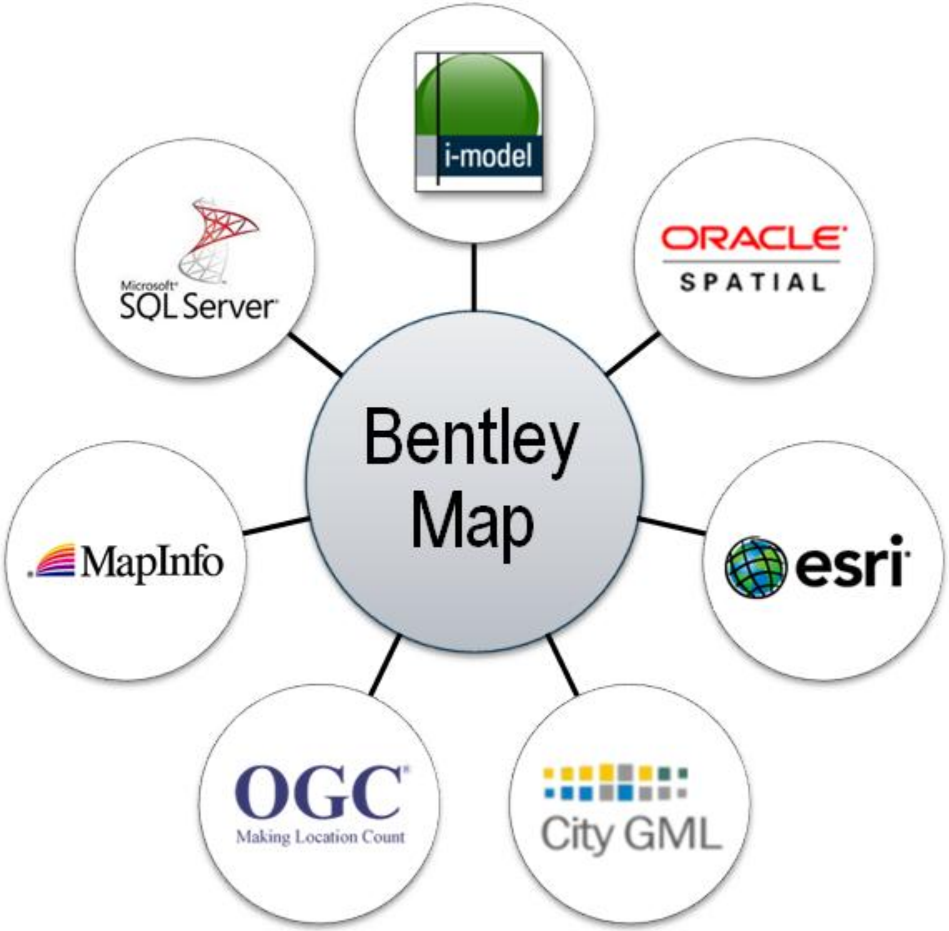
- Adds value to applications by increasing interoperability with other Bentley geospatial products as well as additional data formats.



Frequently Asked Questions

- **Question** - I understand that Bentley Map provides improved interoperability with a number of industry standard data formats beyond those currently supported by MicroStation. Is this correct?

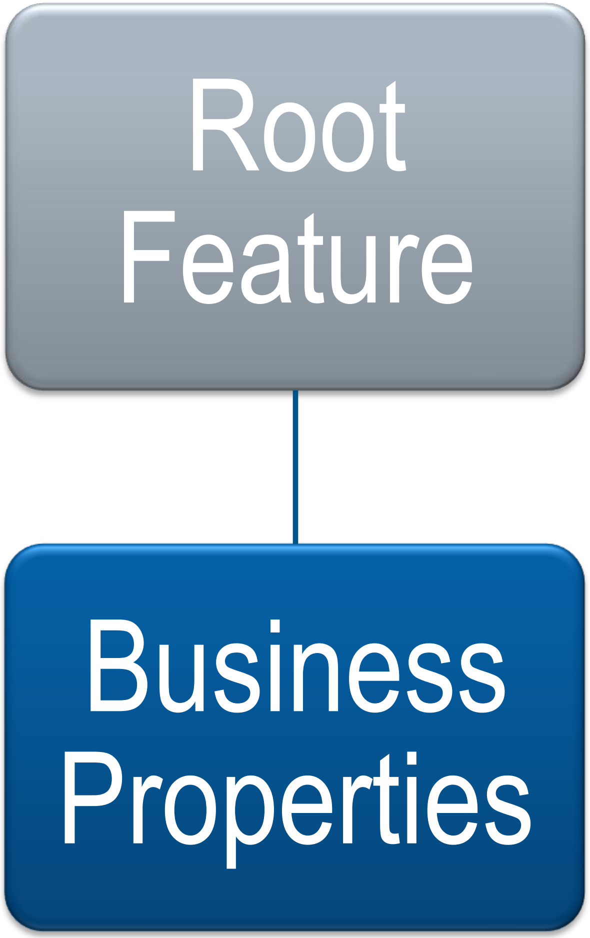
- **Answer** - Yes. From a user and developer perspective, Bentley Map provides native read/write support for many popular industry standard data formats.



Frequently Asked Questions

- **Question** – What is a Bentley Map XFM feature?
- **Answer** – In the simplest terms, a Bentley Map XFM feature is nothing more than one or more MicroStation elements with optionally one or more sets of non-graphic business properties.

Bentley Map XFM Feature Instance



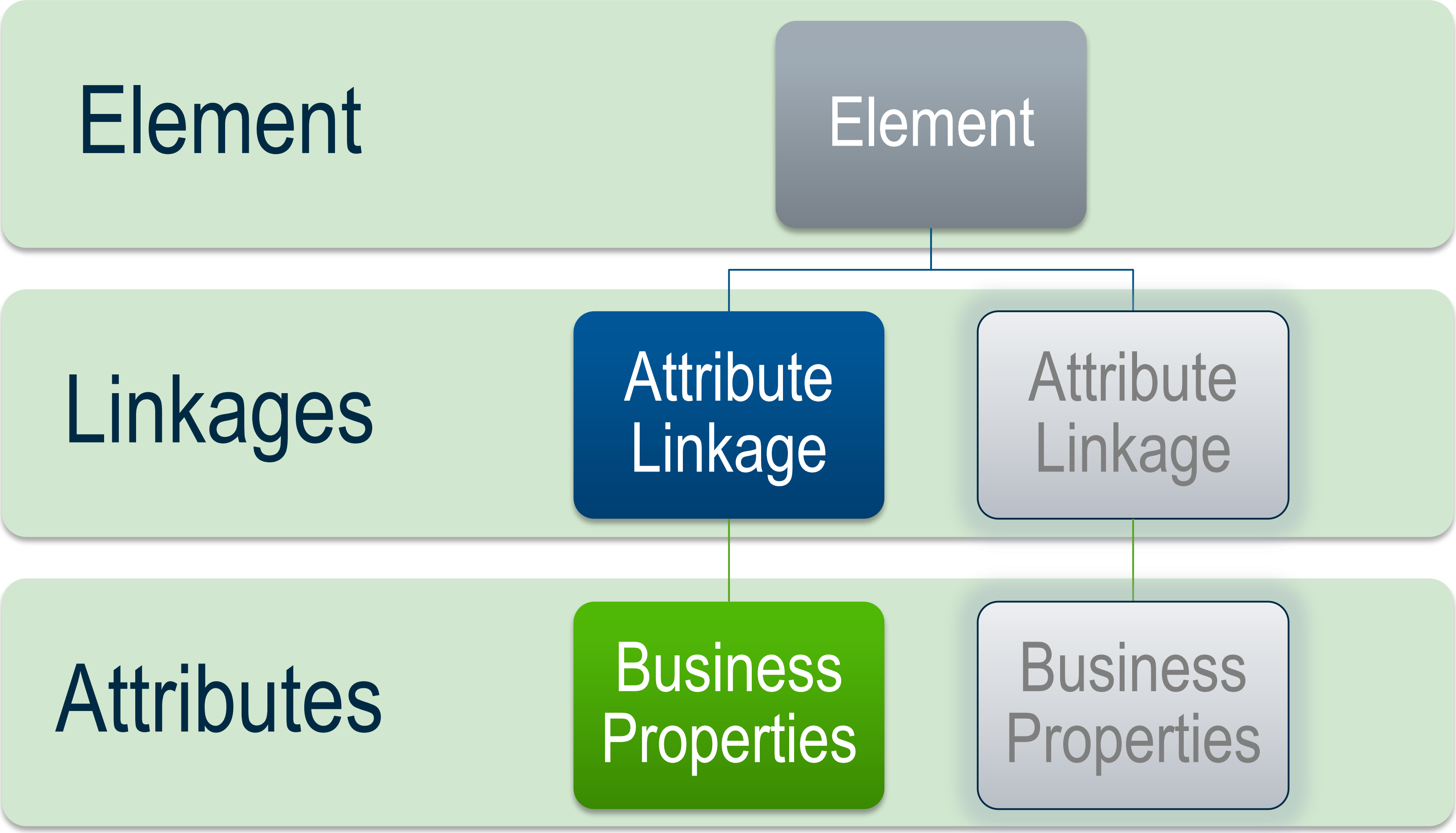
Root Feature

Business Properties

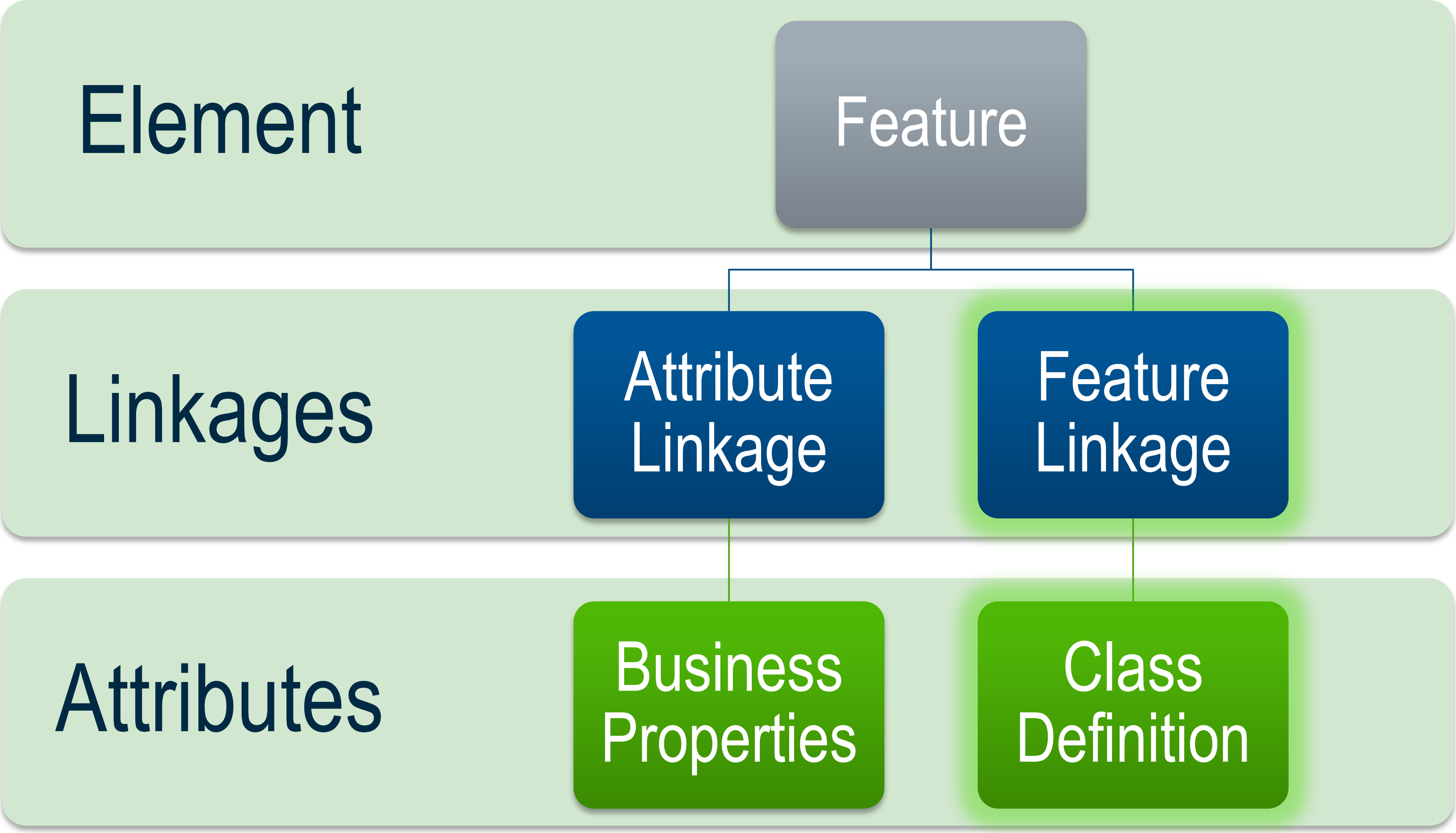
Frequently Asked Questions

- **Question** – How is a Bentley Map XFM feature different than a MicroStation element with a database linkage or a MicroStation GeoGraphics feature?
- **Answer** – While many aspects of an XFM feature instance is similar to a database linked MicroStation element or MicroStation GeoGraphics feature instance there are some unique differences. Let's have a look at the key differences.

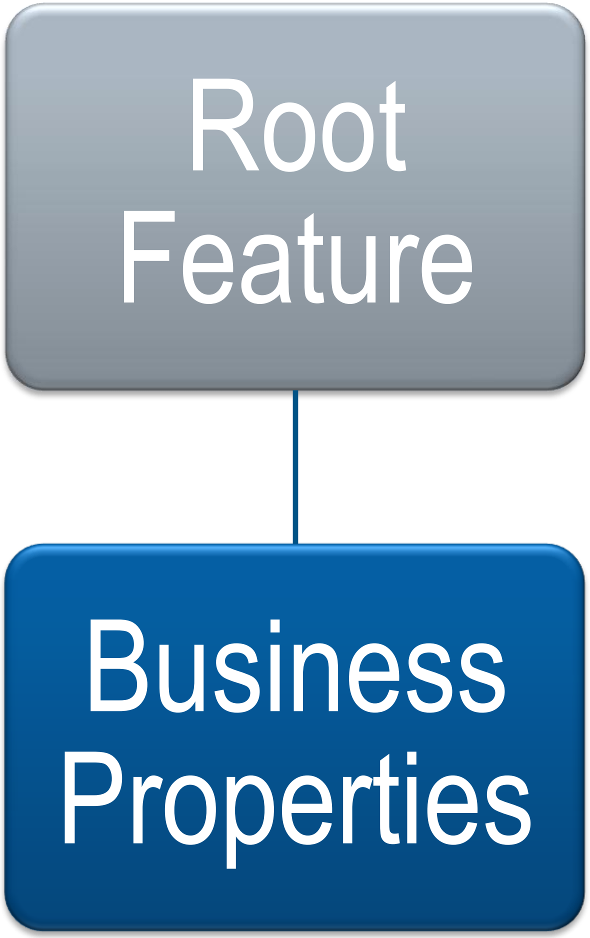
MicroStation Element



MicroStation GeoGraphics Feature Instance



Bentley Map XFM Feature Instance



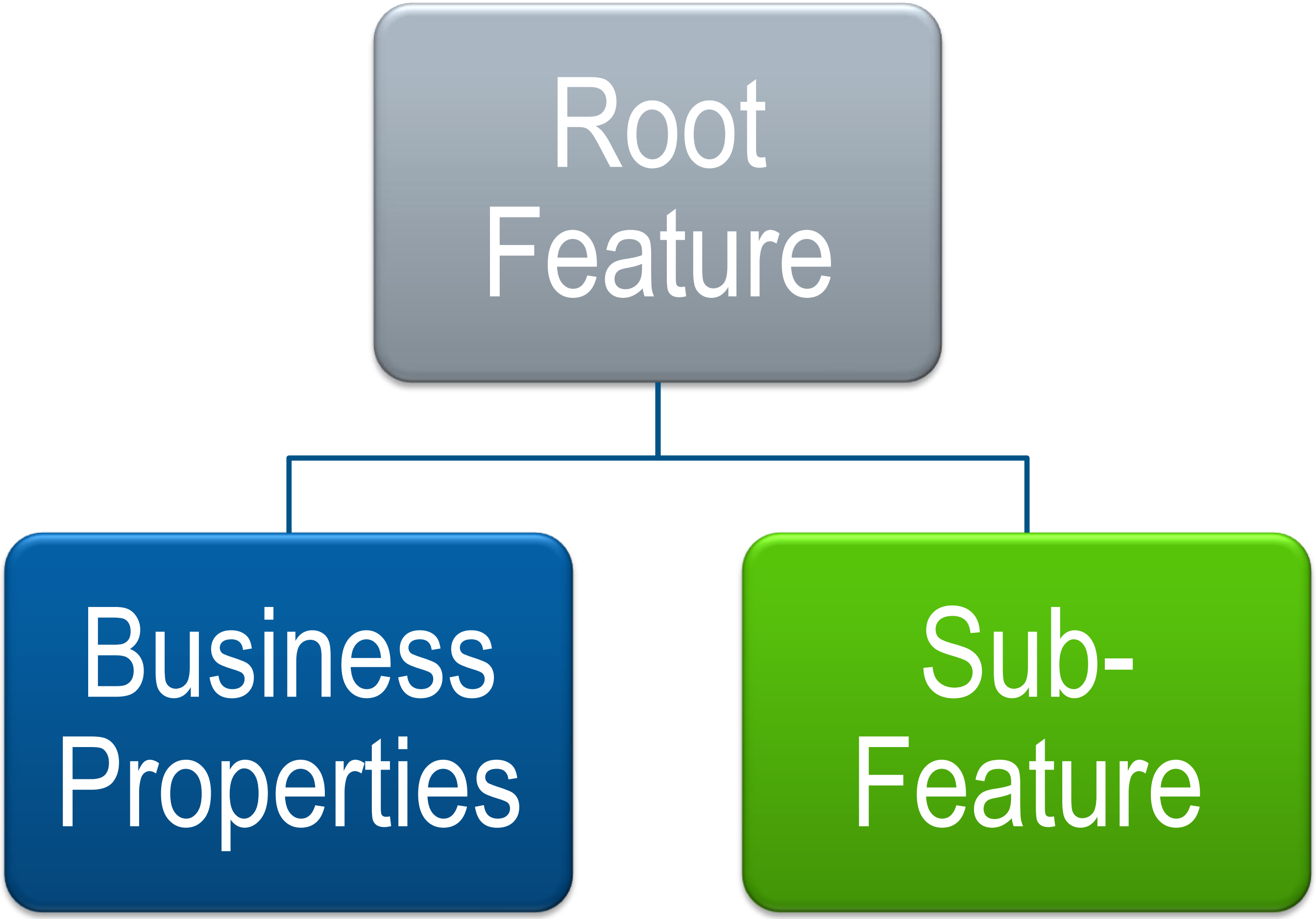
Root Feature

Business Properties

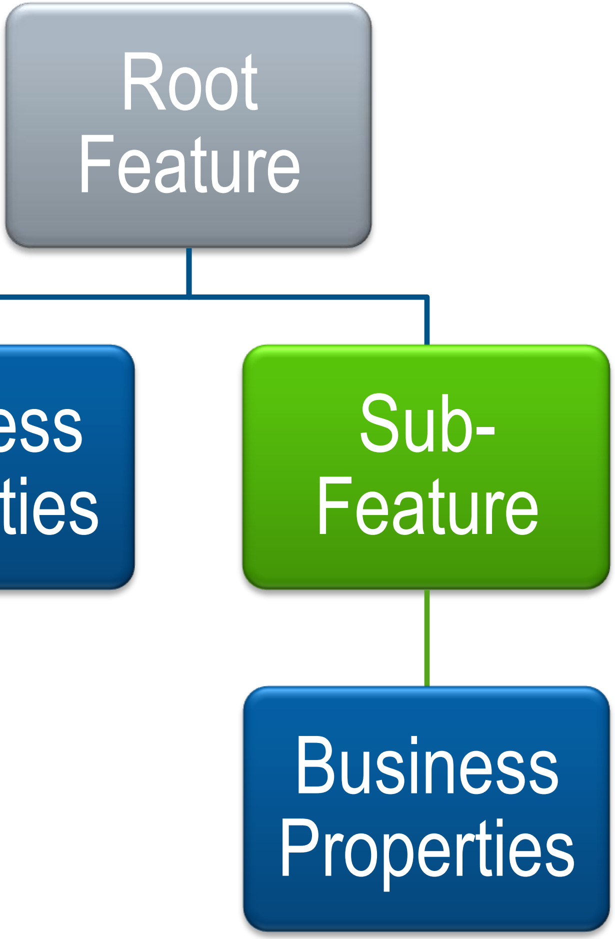
Frequently Asked Questions

- **Question** – What advantages do XFM features give me over regular MicroStation elements?
- **Answer** – As previously stated, a Bentley Map XFM feature is nothing more than one or more MicroStation elements with optionally one or more sets of non-graphic business properties.

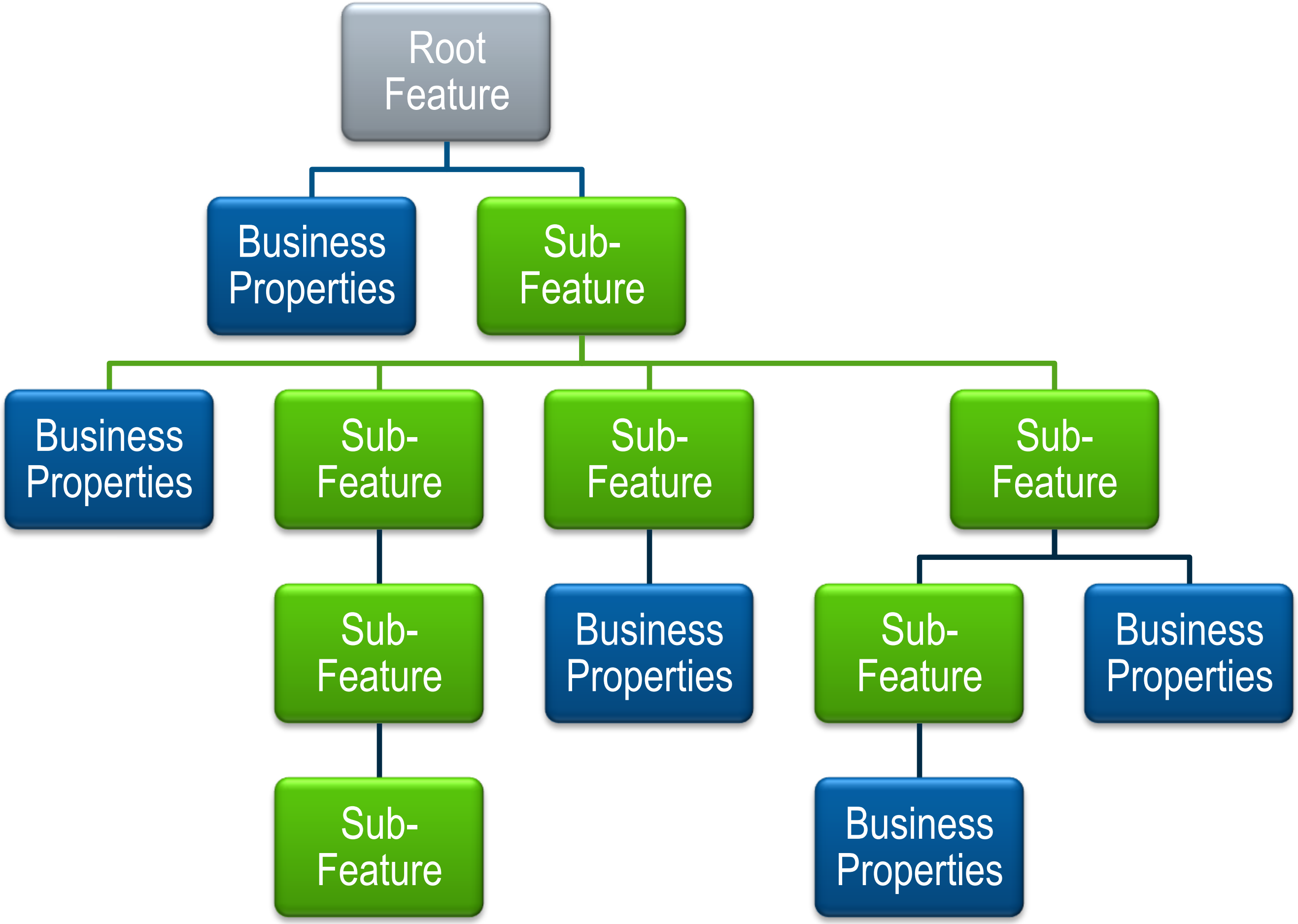
Bentley Map XFM Feature Instance



Bentley Map XFM Feature Instance



Bentley Map XFM Feature Instance



Frequently Asked Questions

- **Question** – Can I use Bentley Map without converting my MicroStation elements to XFM feature instances?
- **Answer** – Yes, using Dynamic Feature Scoring (DFS) technology, Bentley Map has the ability to automatically infer feature classes based upon various aspects of a MicroStation element such as level name and element type.

Frequently Asked Questions

- **Question** – Can I still use MicroStation commands to manipulate XFM feature instances?
- **Answer** – Yes. Bentley Map has been designed in such a way that allows users to take full advantage of the powerful MicroStation editing commands, while at the same time ensuring maintaining referential integrity between the graphic and non-graphic business properties.

Frequently Asked Questions

- **Question** – Are there any existing tools which can help me get started converting my existing MicroStation data to the Bentley Map XFM format?
- **Answer** – Yes, available upon request there exists an example VBA tool called “XFMize” which can be used to convert MicroStation data to the Bentley Map XFM format.

Frequently Asked Questions

- **Question** – What is the Bentley Geospatial Administrator?
- **Answer** – In the simplest terms, the Bentley Geospatial Administrator application is the “class editor” for the Bentley Map platform.

Frequently Asked Questions

- **Question** – Is use of a Bentley Geospatial Administrator generated schema required?
- **Answer** – No. Use of a Bentley Geospatial Administrator generated schema is not required. However there are some additional Bentley Map capabilities which are available whenever the Bentley Geospatial Administrator is used to pre-define a schema, feature classes and their behaviors.

Frequently Asked Questions

- **Question** – What is required to get started with Bentley Map VBA development?
- **Answer** – Bentley Map

Frequently Asked Questions

- **Question** – Which editions of Bentley Map support VBA development?
- **Answer** – All editions including:
  - Bentley Map Enterprise (standalone)
  - Bentley Map (standalone or MicroStation layered)
  - Bentley Map PowerView (standalone)

Bentley
Bentley Map PowerView V8i
(SELECTseries 2)
Bentley's entry-level 3D GIS
Bentley Map V8i
(SELECTseries 2)
Bentley's flagship 3D GIS
Bentley Map Enterprise V8i
(SELECTseries 2)
Bentley's most advanced 3D GIS

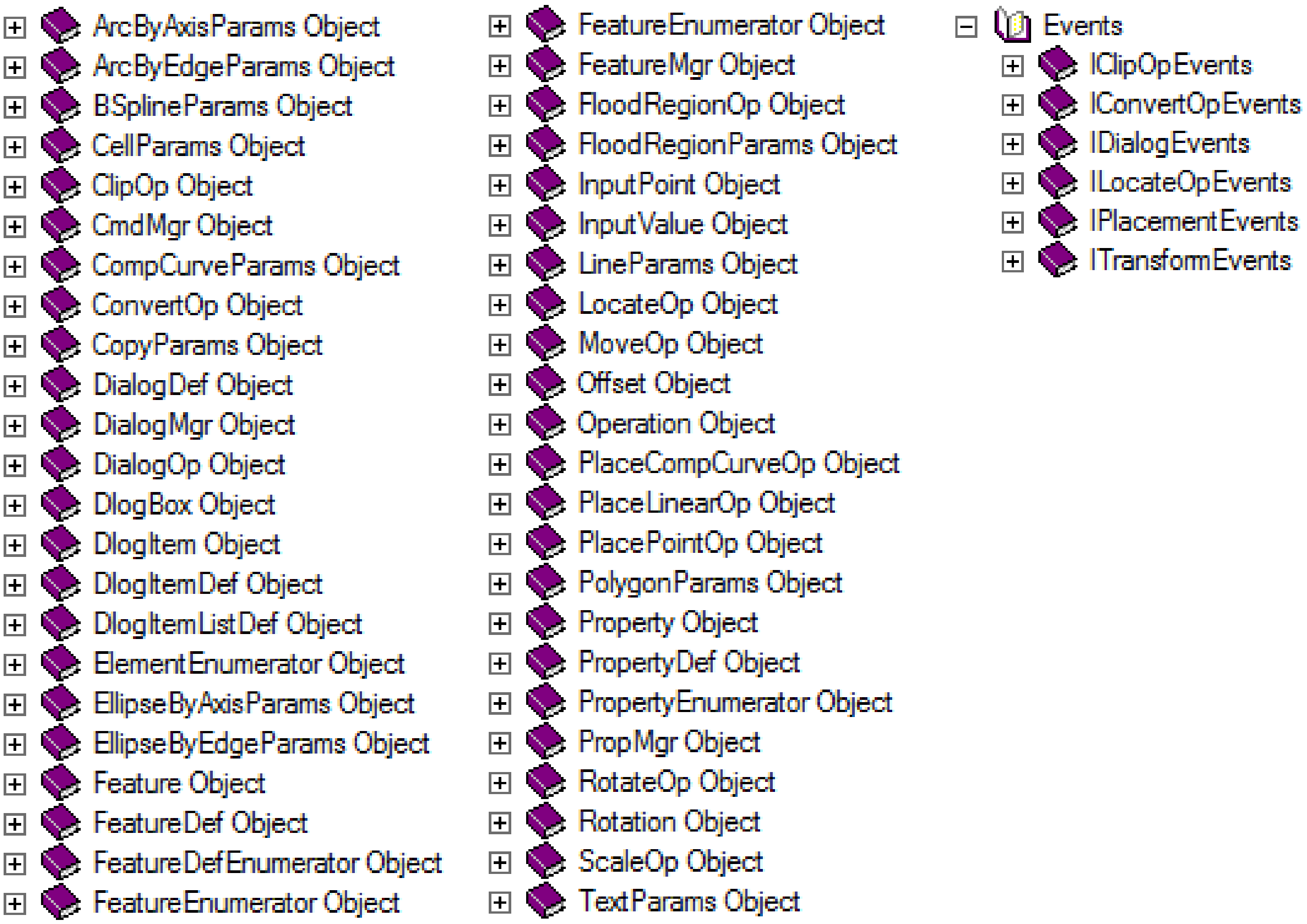
Frequently Asked Questions

- **Question** – What is XFT?
- **Answer** – The term “XFT” stands for “**X**FM **F**eature **T**oolkit” and represents the Component Object Model (COM) for Bentley Map core functionalities.

Frequently Asked Questions

- **Question** – What does the XFT object model provide?
- **Answer** – The XFT object model presents a set of objects and events allowing programmatic access to XFM data modeling and core XFM capabilities. Let’s have a closer look at what the XFT object model provides.

XFT Object Model

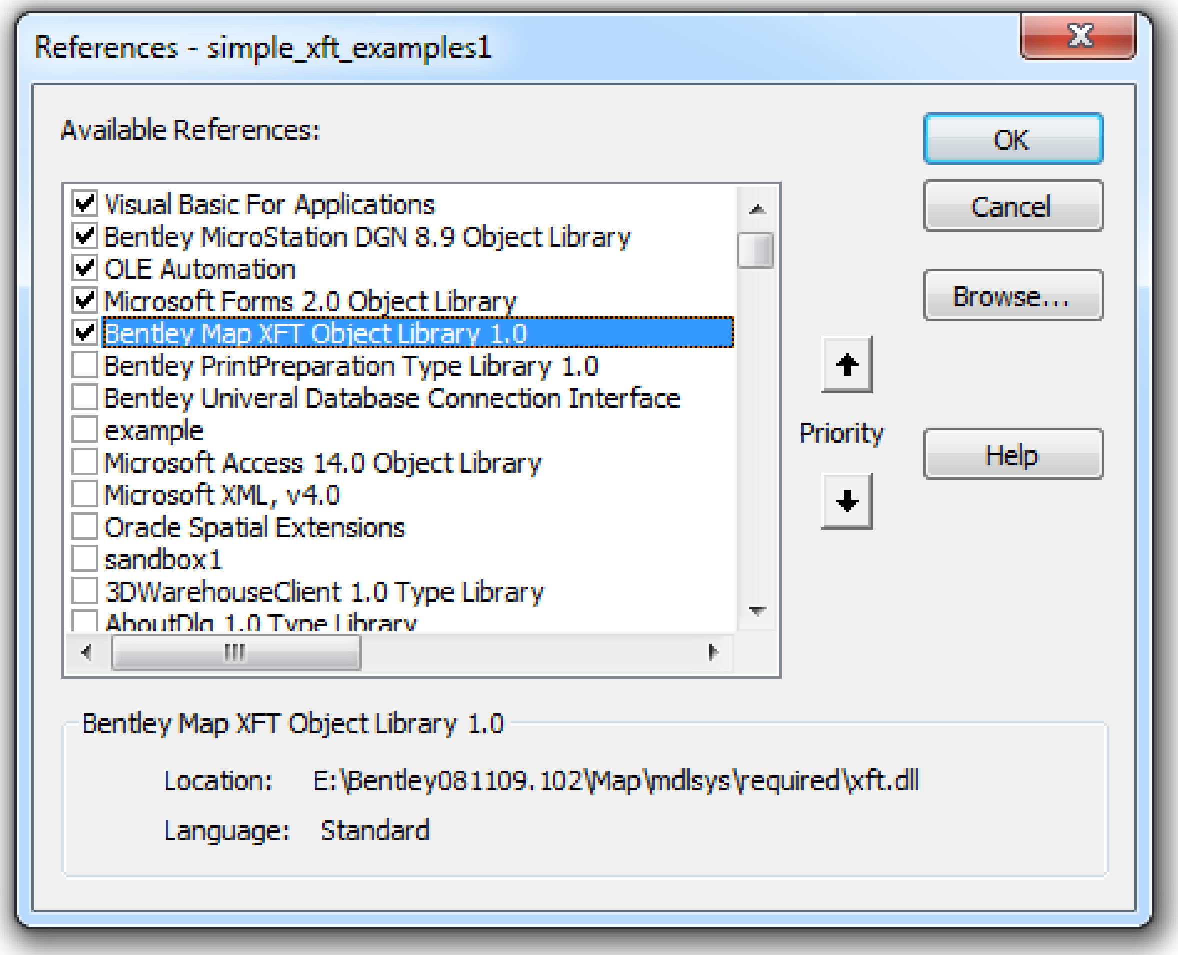


- ⊞ ArcByAxisParams Object
- ⊞ ArcByEdgeParams Object
- ⊞ BSplineParams Object
- ⊞ CellParams Object
- ⊞ ClipOp Object
- ⊞ CmdMgr Object
- ⊞ CompCurveParams Object
- ⊞ ConvertOp Object
- ⊞ CopyParams Object
- ⊞ DialogDef Object
- ⊞ DialogMgr Object
- ⊞ DialogOp Object
- ⊞ DlogBox Object
- ⊞ DlogItem Object
- ⊞ DlogItemDef Object
- ⊞ DlogItemListDef Object
- ⊞ ElementEnumerator Object
- ⊞ EllipseByAxisParams Object
- ⊞ EllipseByEdgeParams Object
- ⊞ Feature Object
- ⊞ FeatureDef Object
- ⊞ FeatureDefEnumerator Object
- ⊞ FeatureEnumerator Object
- ⊞ FeatureEnumerator Object
- ⊞ FeatureMgr Object
- ⊞ FloodRegionOp Object
- ⊞ FloodRegionParams Object
- ⊞ InputPoint Object
- ⊞ InputValue Object
- ⊞ LineParams Object
- ⊞ LocateOp Object
- ⊞ MoveOp Object
- ⊞ Offset Object
- ⊞ Operation Object
- ⊞ PlaceCompCurveOp Object
- ⊞ PlaceLinearOp Object
- ⊞ PlacePointOp Object
- ⊞ PolygonParams Object
- ⊞ Property Object
- ⊞ PropertyDef Object
- ⊞ PropertyEnumerator Object
- ⊞ PropMgr Object
- ⊞ RotateOp Object
- ⊞ Rotation Object
- ⊞ ScaleOp Object
- ⊞ TextParams Object
- ⊟ Events
  - ⊞ IClipOpEvents
  - ⊞ IConvertOpEvents
  - ⊞ IDialogEvents
  - ⊞ ILocateOpEvents
  - ⊞ IPlacementEvents
  - ⊞ ITransformEvents

Frequently Asked Questions

- **Question** – So now that I understand a bit more about XFM data modeling and the XFT object model, are Bentley Map VBA applications difficult to write?
- **Answer** – No, not really. Using Bentley Map VBA, basic operations such as creating features and sub-features, adding business properties, setting symbology and locating features are actually quite simple. Let’s have a look at a few brief examples.

VBA References



References - simple_xft_examples1

Available References:

- ☑ Visual Basic For Applications
- ☑ Bentley MicroStation DGN 8.9 Object Library
- ☑ OLE Automation
- ☑ Microsoft Forms 2.0 Object Library
- ☑ Bentley Map XFT Object Library 1.0
- □ Bentley PrintPreparation Type Library 1.0
- □ Bentley Univeral Database Connection Interface
- □ example
- □ Microsoft Access 14.0 Object Library
- □ Microsoft XML, v4.0
- □ Oracle Spatial Extensions
- □ sandbox1
- □ 3DWarehouseClient 1.0 Type Library
- □ AboutDlg 1.0 Type Library

Bentley Map XFT Object Library 1.0

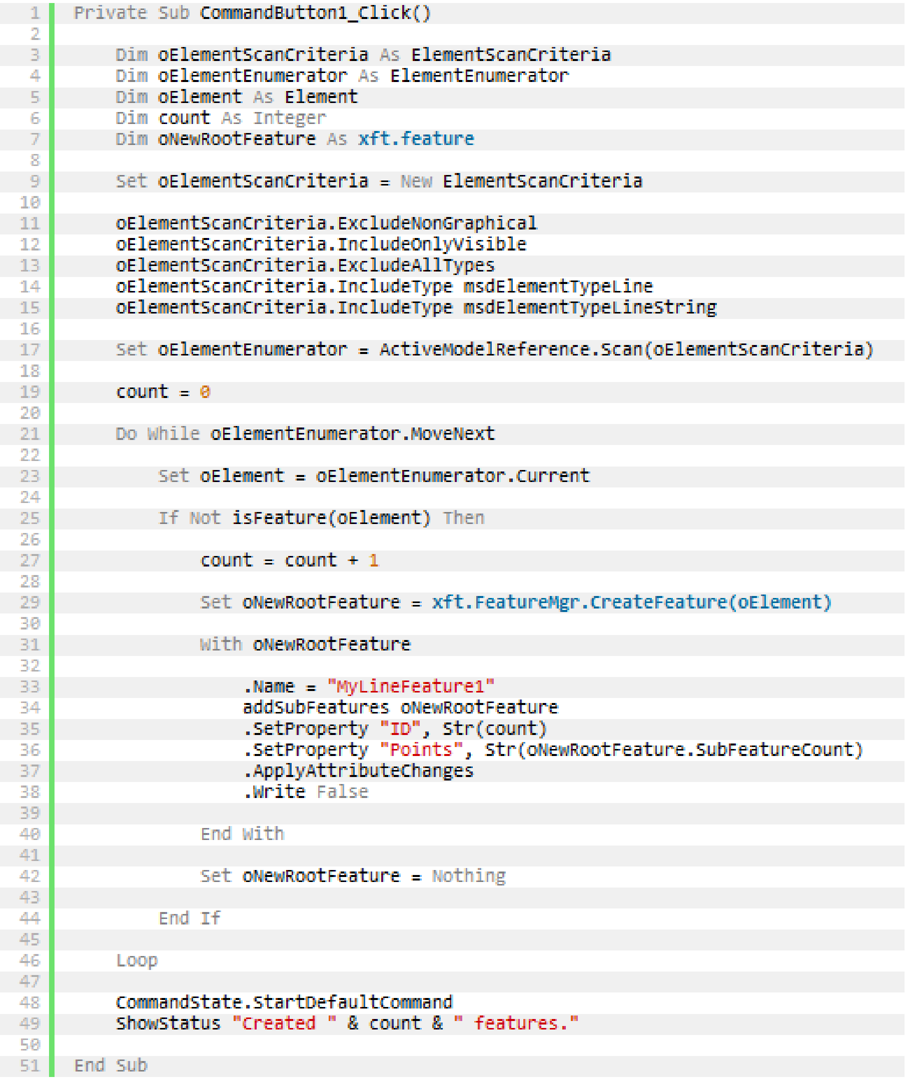
Location: E:\Bentley081109.102\Map\mdlsys\required\xft.dll

Language: Standard

Bentley Map VBA Example

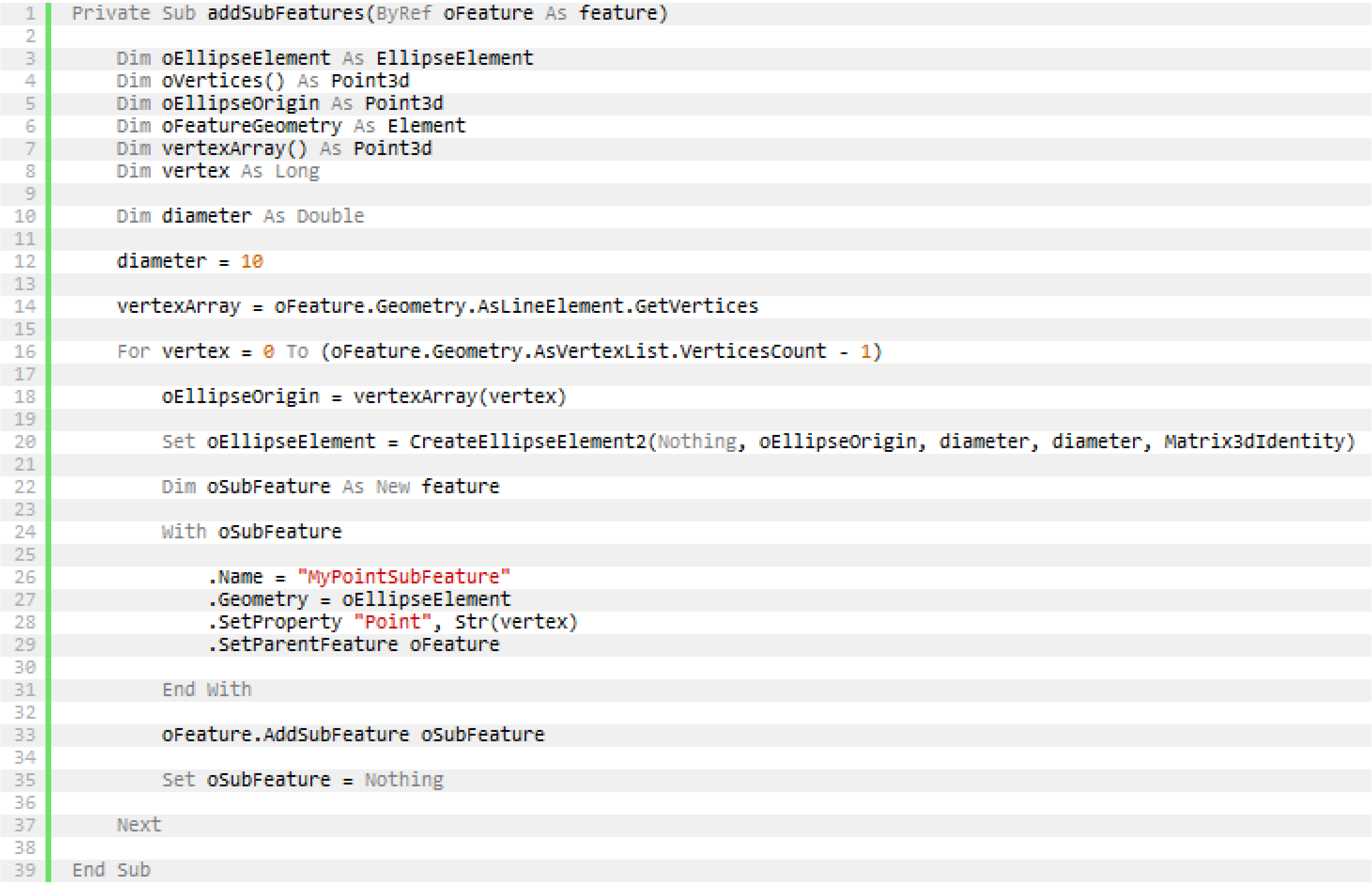
- Find MicroStation line or linestring elements and convert them to XFM feature instances.
- Create new elements and add those as sub-features.
- Add business properties to both the root and all sub-features.
- Create a locate operation that searches the entire design file, fence contents or selection set for feature instances matching a particular search criteria.
- Load search results into the Data Browser application.

Create Feature Instances



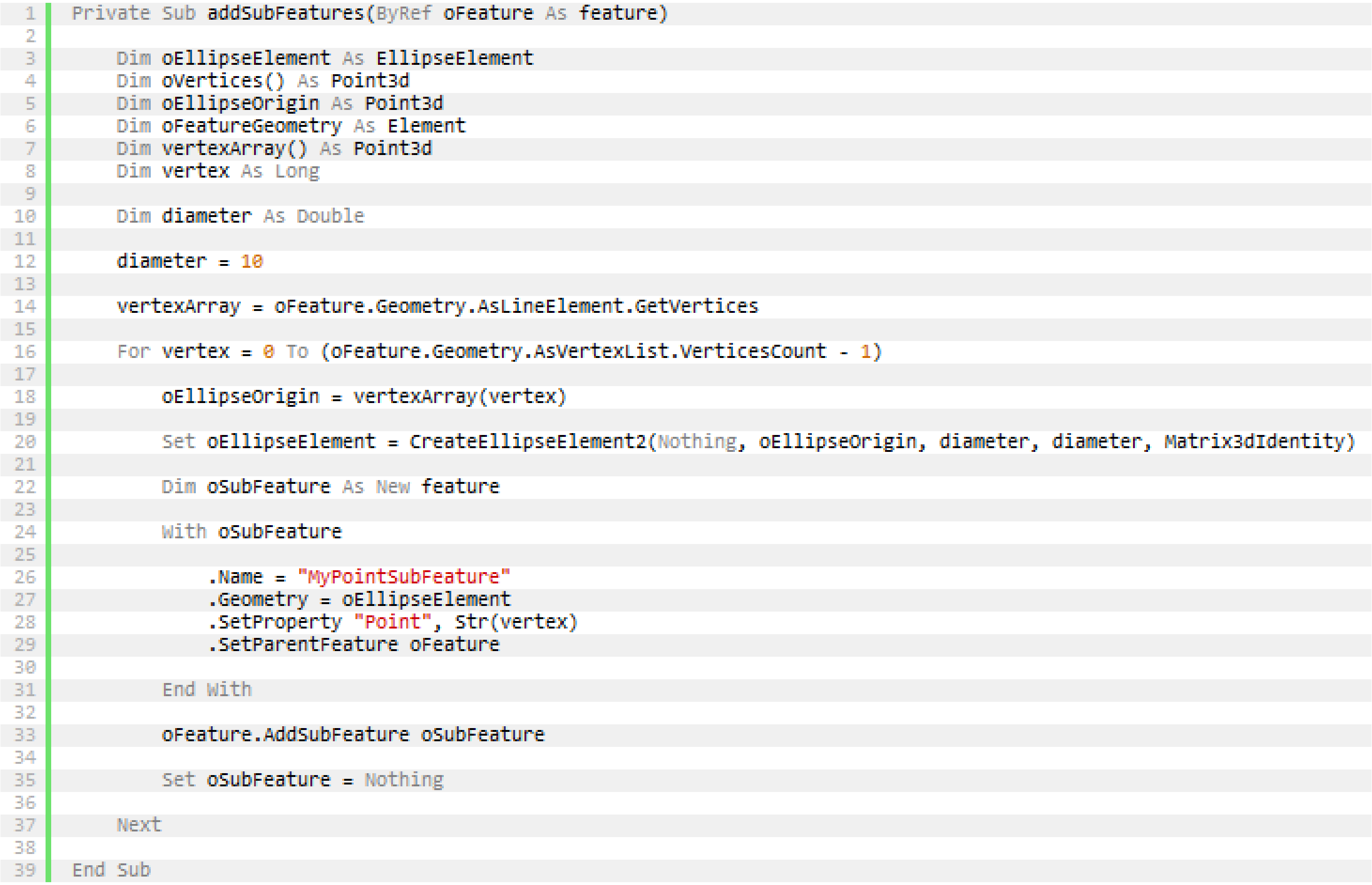
```vb
Private Sub CommandButton1_Click()

    Dim oElementScanCriteria As ElementScanCriteria
    Dim oElementEnumerator As ElementEnumerator
    Dim oElement As Element
    Dim count As Integer
    Dim oNewRootFeature As xft.feature

    Set oElementScanCriteria = New ElementScanCriteria

    oElementScanCriteria.ExcludeNonGraphical
    oElementScanCriteria.IncludeOnlyVisible
    oElementScanCriteria.ExcludeAllTypes
    oElementScanCriteria.IncludeType msdElementTypeLine
    oElementScanCriteria.IncludeType msdElementTypeLineString

    Set oElementEnumerator = ActiveModelReference.Scan(oElementScanCriteria)

    count = 0

    Do While oElementEnumerator.MoveNext

        Set oElement = oElementEnumerator.Current

        If Not isFeature(oElement) Then

            count = count + 1

            Set oNewRootFeature = xft.FeatureMgr.CreateFeature(oElement)

            With oNewRootFeature

                .Name = "MyLineFeature1"
                addSubFeatures oNewRootFeature
                .SetProperty "ID", Str(count)
                .SetProperty "Points", Str(oNewRootFeature.SubFeatureCount)
                .ApplyAttributeChanges
                .Write False

            End With

            Set oNewRootFeature = Nothing

        End If

    Loop

    CommandState.StartDefaultCommand
    ShowStatus "Created " & count & " features."

End Sub
```

Create Sub-Features



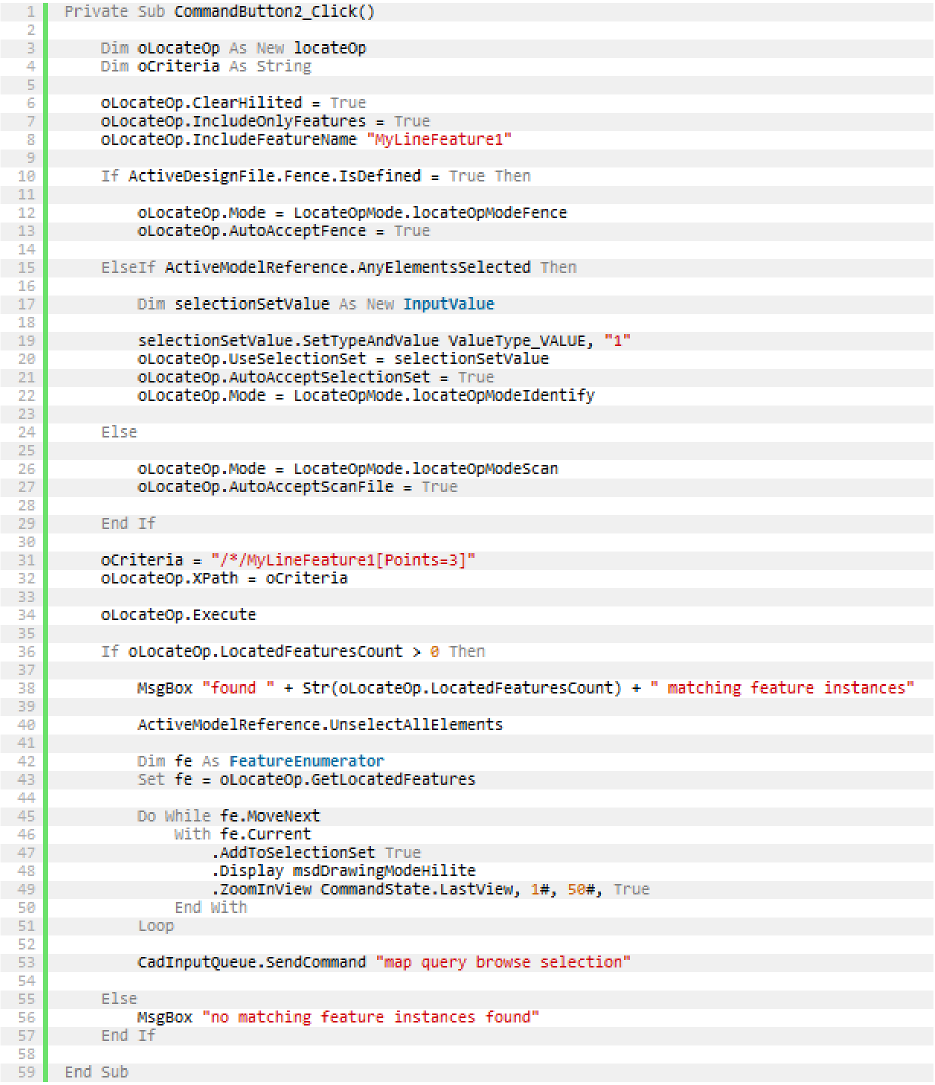
```
Private Sub addSubFeatures(ByRef oFeature As feature)

    Dim oEllipseElement As EllipseElement
    Dim oVertices() As Point3d
    Dim oEllipseOrigin As Point3d
    Dim oFeatureGeometry As Element
    Dim vertexArray() As Point3d
    Dim vertex As Long

    Dim diameter As Double

    diameter = 10

    vertexArray = oFeature.Geometry.AsLineElement.GetVertices

    For vertex = 0 To (oFeature.Geometry.AsVertexList.VerticesCount - 1)

        oEllipseOrigin = vertexArray(vertex)

        Set oEllipseElement = CreateEllipseElement2(Nothing, oEllipseOrigin, diameter, diameter, Matrix3dIdentity)

        Dim oSubFeature As New feature

        With oSubFeature

            .Name = "MyPointSubFeature"
            .Geometry = oEllipseElement
            .SetProperty "Point", Str(vertex)
            .SetParentFeature oFeature

        End With

        oFeature.AddSubFeature oSubFeature

        Set oSubFeature = Nothing

    Next

End Sub
```

Add Business Properties



```
Private Sub addSubFeatures(ByRef oFeature As feature)

    Dim oEllipseElement As EllipseElement
    Dim oVertices() As Point3d
    Dim oEllipseOrigin As Point3d
    Dim oFeatureGeometry As Element
    Dim vertexArray() As Point3d
    Dim vertex As Long

    Dim diameter As Double

    diameter = 10

    vertexArray = oFeature.Geometry.AsLineElement.GetVertices

    For vertex = 0 To (oFeature.Geometry.AsVertexList.VerticesCount - 1)

        oEllipseOrigin = vertexArray(vertex)

        Set oEllipseElement = CreateEllipseElement2(Nothing, oEllipseOrigin, diameter, diameter, Matrix3dIdentity)

        Dim oSubFeature As New feature

        With oSubFeature

            .Name = "MyPointSubFeature"
            .Geometry = oEllipseElement
            .SetProperty "Point", Str(vertex)
            .SetParentFeature oFeature

        End With

        oFeature.AddSubFeature oSubFeature

        Set oSubFeature = Nothing

    Next

End Sub
```

Locate Feature Instances



```
Private Sub CommandButton2_Click()

    Dim oLocateOp As New locateOp
    Dim oCriteria As String

    oLocateOp.ClearHilited = True
    oLocateOp.IncludeOnlyFeatures = True
    oLocateOp.IncludeFeatureName "MyLineFeature1"

    If ActiveDesignFile.Fence.IsDefined = True Then

        oLocateOp.Mode = LocateOpMode.locateOpModeFence
        oLocateOp.AutoAcceptFence = True

    ElseIf ActiveModelReference.AnyElementsSelected Then

        Dim selectionSetValue As New InputValue

        selectionSetValue.SetTypeAndValue ValueType_VALUE, "1"
        oLocateOp.UseSelectionSet = selectionSetValue
        oLocateOp.AutoAcceptSelectionSet = True
        oLocateOp.Mode = LocateOpMode.locateOpModeIdentify

    Else

        oLocateOp.Mode = LocateOpMode.locateOpModeScan
        oLocateOp.AutoAcceptScanFile = True

    End If

    oCriteria = "/*/MyLineFeature1[Points=3]"
    oLocateOp.XPath = oCriteria

    oLocateOp.Execute

    If oLocateOp.LocatedFeaturesCount > 0 Then

        MsgBox "found " + Str(oLocateOp.LocatedFeaturesCount) + " matching feature instances"

        ActiveModelReference.UnselectAllElements

        Dim fe As FeatureEnumerator
        Set fe = oLocateOp.GetLocatedFeatures

        Do While fe.MoveNext
            With fe.Current
                .AddToSelectionSet True
                .Display msdDrawingModeHilite
                .ZoomInView CommandState.LastView, 1#, 50#, True
            End With
        Loop

        CadInputQueue.SendCommand "map query browse selection"

    Else
        MsgBox "no matching feature instances found"
    End If

End Sub
```

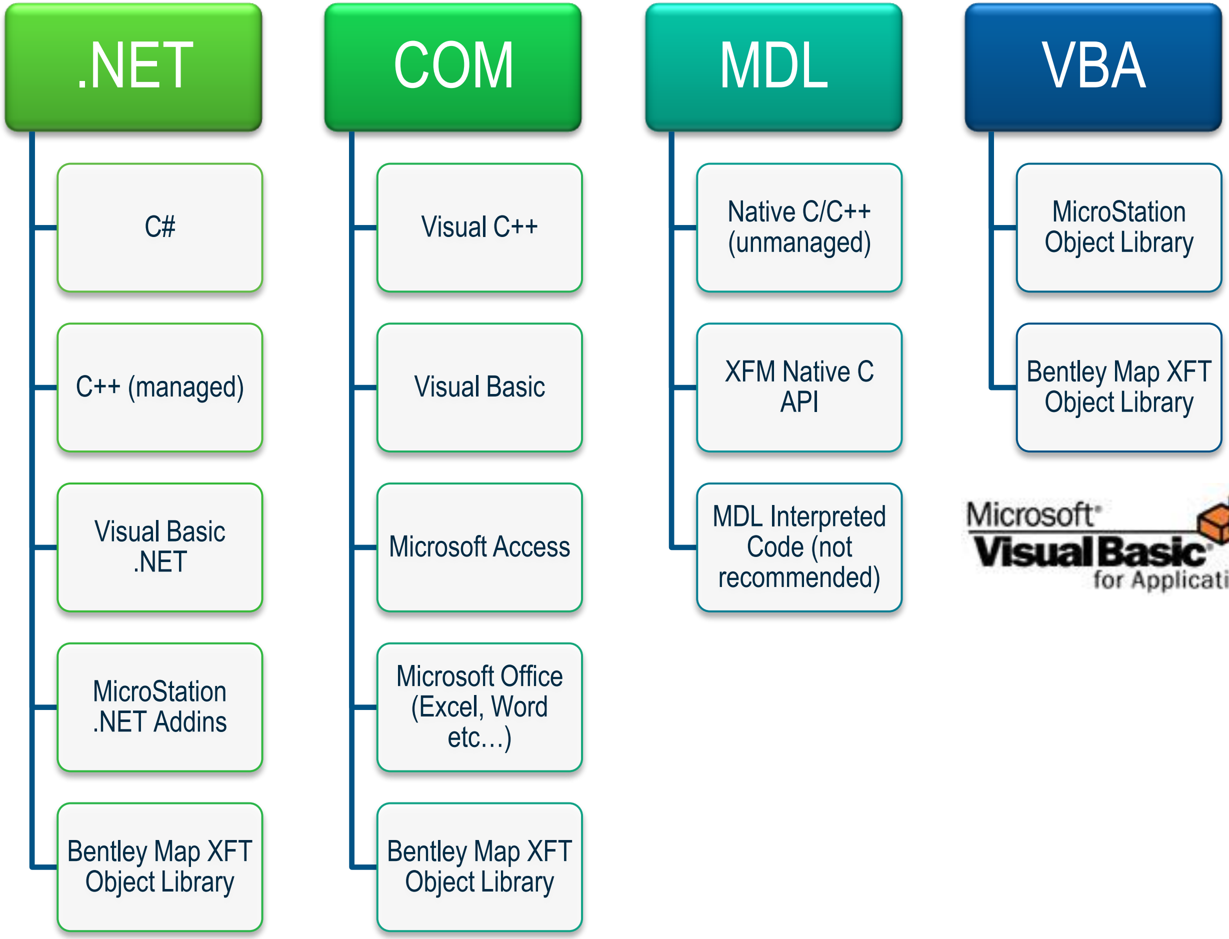
Frequently Asked Questions

- **Question** – Can I use MicroStation VBA with Bentley Map VBA?
- **Answer** – Yes and in some cases use of MicroStation VBA is required. For example, when working with the geometry of an XFM feature instance, element creation or manipulations are done using standard MicroStation VBA objects and methods.

Frequently Asked Questions

- **Question** – By learning Bentley Map development using VBA and the XFT object model, what advantages do I gain?
- **Answer** – Beyond having the ability to create VBA applications to create, manipulate and analyze Bentley Map XFM data, by learning the XFT object model a transition to .NET development using C# or Visual Basic .NET becomes easier since XFT can also be used with COM interop.

Development Languages



.NET

- C#
- C++ (managed)
- Visual Basic .NET
- MicroStation .NET Addins
- Bentley Map XFT Object Library

COM

- Visual C++
- Visual Basic
- Microsoft Access
- Microsoft Office (Excel, Word etc…)
- Bentley Map XFT Object Library

MDL

- Native C/C++ (unmanaged)
- XFM Native C API
- MDL Interpreted Code (not recommended)

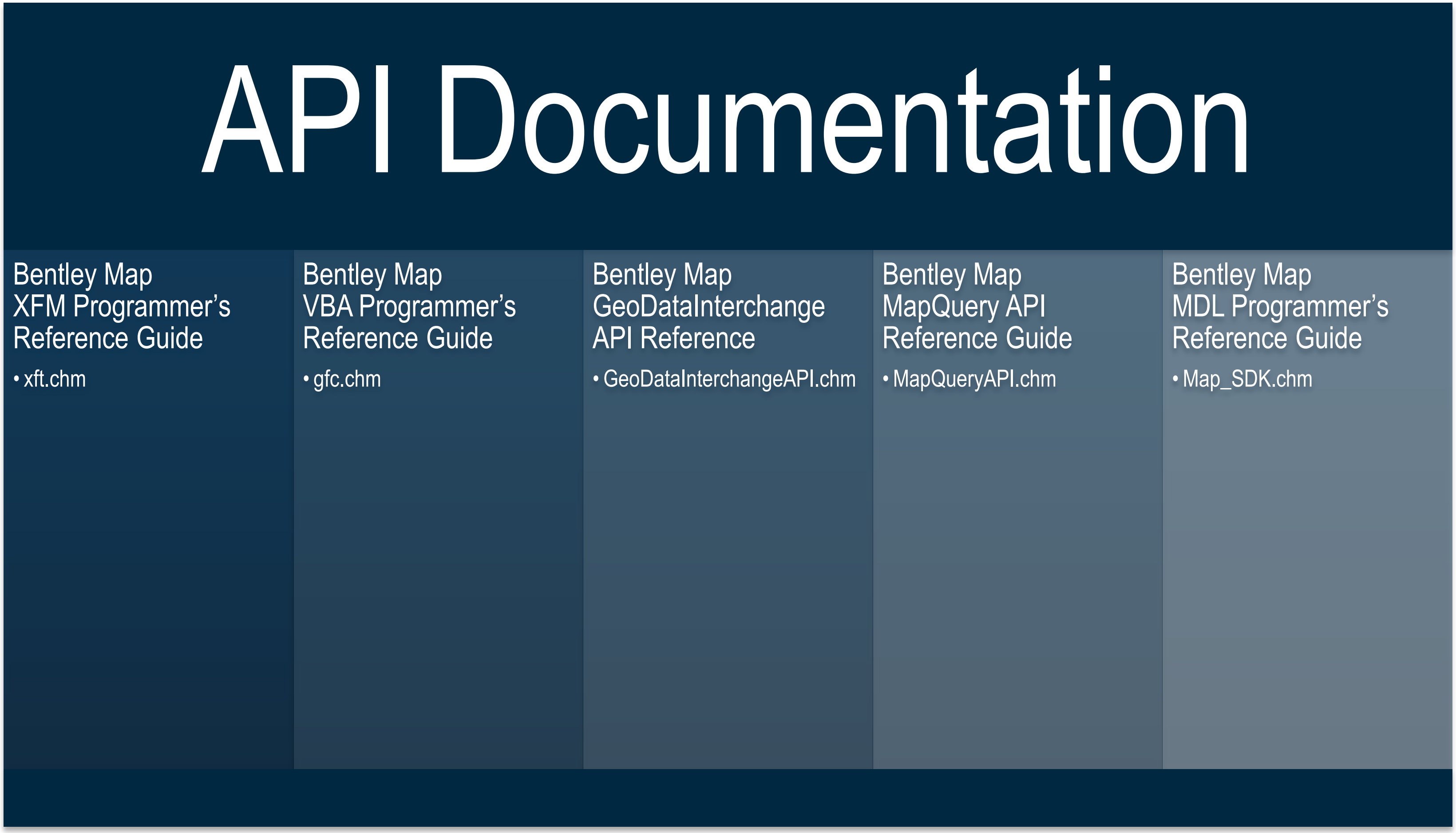
VBA

- MicroStation Object Library
- Bentley Map XFT Object Library

Bentley Map

Additional Resources

Documentation



API Documentation

Bentley Map XFM Programmer’s Reference Guide

- xft.chm

Bentley Map VBA Programmer’s Reference Guide

- gfc.chm

Bentley Map GeoDataInterchange API Reference

- GeoDataInterchangeAPI.chm

Bentley Map MapQuery API Reference Guide

- MapQueryAPI.chm

Bentley Map MDL Programmer’s Reference Guide

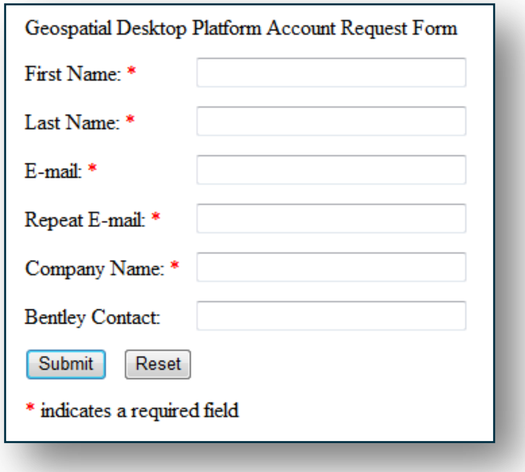
- Map_SDK.chm

Geospatial Desktop Platform Extranet

Bentley Geospatial Desktop Platform Extranet

RATE THIS ☆☆☆☆☆

The "Bentley Geospatial Desktop Platform" extranet site hosts the "**Bentley Map - Development and Product Customization Guide**" and "**Bentley Map - Product Development Notebooks**" which are frequently updated online resources for anyone interested in learning more about customizing or developing applications for the Bentley Map product line.



Geospatial Desktop Platform Account Request Form

First Name: *	
Last Name: *	
E-mail: *	
Repeat E-mail: *	
Company Name: *	
Bentley Contact:	

Submit Reset

* indicates a required field

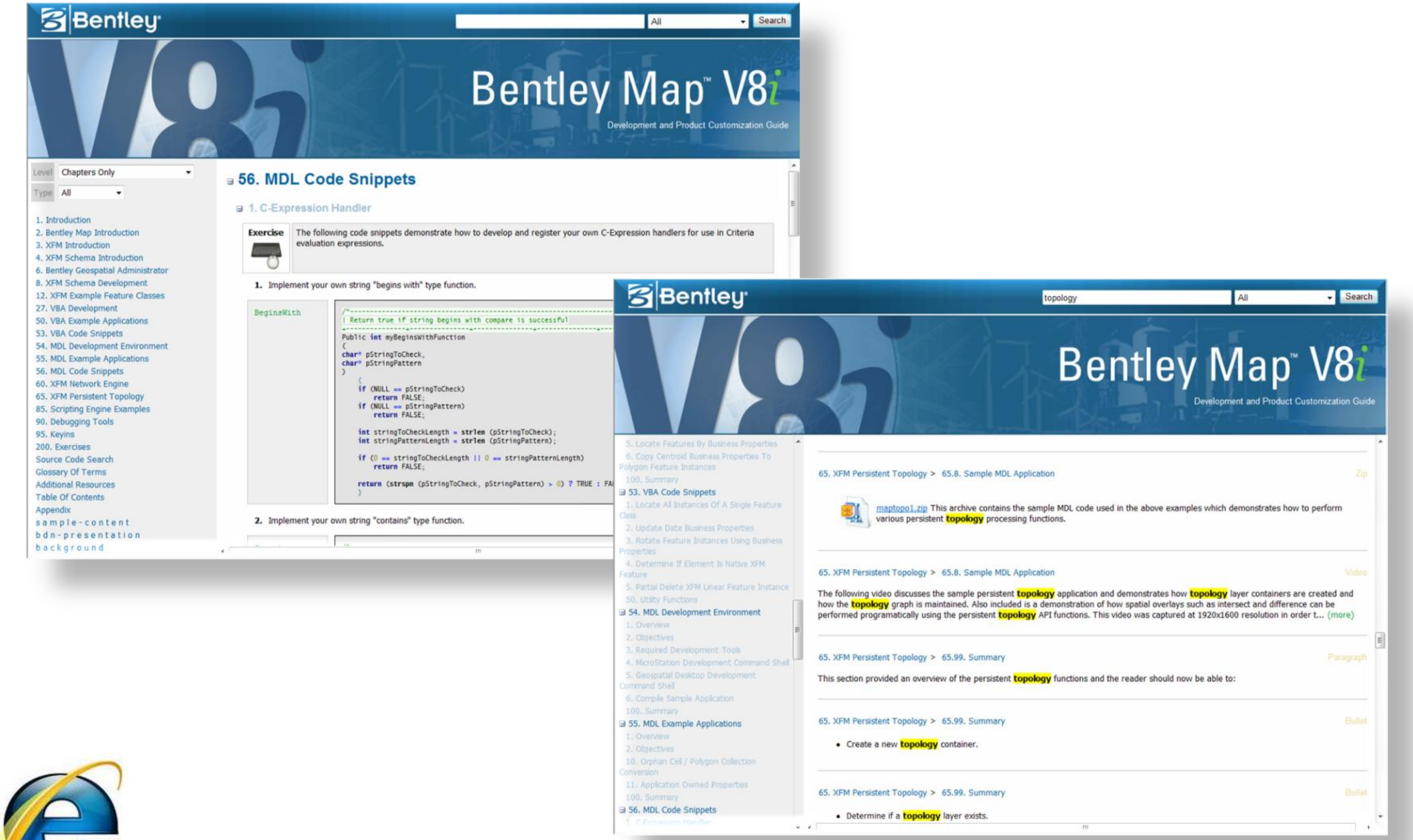
If your organization is a current Bentley SELECT subscriber or Bentley Development Network (BDN) member you may request access to these online resources using the **Geospatial Desktop Platform Account Request Form** listing your Account Manager (preferred) or Jeff Bielefeld as the Bentley contact.

Please note that access requests may take few a days to be processed. Once approved you will receive a confirmation mail message with additional information. Access requests submitted with incomplete information or those not including a valid and verifiable Bentley contact cannot be approved.

If you have any questions regarding the above, please contact Jeff Bielefeld at Bentley for more information.

☆ Favorite this page | Bentley Map, Development, Programming, XFM, XFT Edit tags

Online Development Guide



1. Introduction
2. Bentley Map Introduction
3. XFM Introduction
4. XFM Schema Introduction
6. Bentley Geospatial Administrator
8. XFM Schema Development
12. XFM Example Feature Classes
27. VBA Development
50. VBA Example Applications
53. VBA Code Snippets
54. MDL Development Environment
55. MDL Example Applications
56. MDL Code Snippets
60. XFM Network Engine
65. XFM Persistent Topology
85. Scripting Engine Examples
90. Debugging Tools
95. Keyins
200. Exercises

Source Code Search
Glossary Of Terms
Additional Resources
Table Of Contents
Appendix
sample-content
bdn-presentation
background

56. MDL Code Snippets

1. C-Expression Handler

Exercise The following code snippets demonstrate how to develop and register your own C-Expression handlers for use in Criteria evaluation expressions.

1. Implement your own string "begins with" type function.

BeginsWith

```
/*----------------------------------------------------------------
| Return true if string begins with compare is successful
+-------------------------------------------------------------+--
Public int myBeginsWithFunction
(
char* pStringToCheck,
char* pStringPattern
)
    {
    if (NULL == pStringToCheck)
        return FALSE;
    if (NULL == pStringPattern)
        return FALSE;

    int stringToCheckLength = strlen (pStringToCheck);
    int stringPatternLength = strlen (pStringPattern);

    if (0 == stringToCheckLength || 0 == stringPatternLength)
        return FALSE;

    return (strspn (pStringToCheck, pStringPattern) > 0) ? TRUE : FA
    }
```

2. Implement your own string "contains" type function.

5. Locate Features By Business Properties
6. Copy Centroid Business Properties To Polygon Feature Instances
100. Summary

53. VBA Code Snippets
1. Locate All Instances Of A Single Feature Class
2. Update Date Business Properties
3. Rotate Feature Instances Using Business Properties
4. Determine If Element Is Native XFM Feature
5. Partial Delete XFM Linear Feature Instance
50. Utility Functions

54. MDL Development Environment
1. Overview
2. Objectives
3. Required Development Tools
4. MicroStation Development Command Shell
5. Geospatial Desktop Development Command Shell
6. Compile Sample Application
100. Summary

55. MDL Example Applications
1. Overview
2. Objectives
10. Orphan Cell / Polygon Collection Conversion
11. Application Owned Properties
100. Summary

56. MDL Code Snippets
1. C-Expression Handler

65. XFM Persistent Topology > 65.8. Sample MDL Application — Zip

maptopo1.zip This archive contains the sample MDL code used in the above examples which demonstrates how to perform various persistent topology processing functions.

65. XFM Persistent Topology > 65.8. Sample MDL Application — Video

The following video discusses the sample persistent topology application and demonstrates how topology layer containers are created and how the topology graph is maintained. Also included is a demonstration of how spatial overlays such as intersect and difference can be performed programatically using the persistent topology API functions. This video was captured at 1920x1600 resolution in order t... (more)

65. XFM Persistent Topology > 65.99. Summary — Paragraph

This section provided an overview of the persistent topology functions and the reader should now be able to:

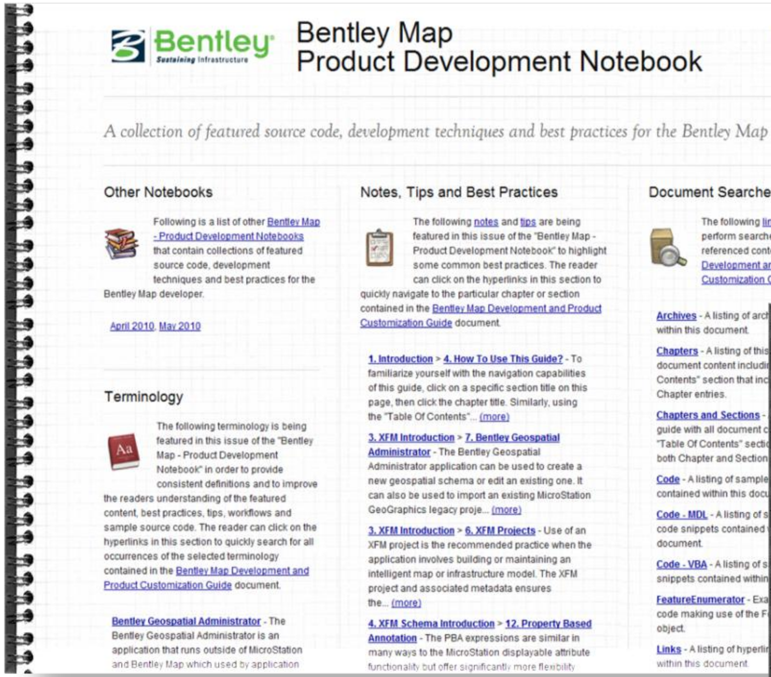
65. XFM Persistent Topology > 65.99. Summary — Bullet

- Create a new topology container.

65. XFM Persistent Topology > 65.99. Summary — Bullet

- Determine if a topology layer exists.

Online Developer Notebooks



Bentley Map
Product Development Notebook

March 2010

A collection of featured source code, development techniques and best practices for the Bentley Map developer.

Other Notebooks

Following is a list of other Bentley Map - Product Development Notebooks that contain collections of featured source code, development techniques and best practices for the Bentley Map developer.

April 2010, May 2010

Terminology

The following terminology is being featured in this issue of the "Bentley Map - Product Development Notebook" in order to provide consistent definitions and to improve the readers understanding of the featured content, best practices, tips, workflows and sample source code. The reader can click on the hyperlinks in this section to quickly search for all occurrences of the selected terminology contained in the Bentley Map Development and Product Customization Guide document.

Bentley Geospatial Administrator - The Bentley Geospatial Administrator is an application that runs outside of MicroStation and Bentley Map which used by application

Notes, Tips and Best Practices

The following notes and tips are being featured in this issue of the "Bentley Map - Product Development Notebook" to highlight some common best practices. The reader can click on the hyperlinks in this section to quickly navigate to the particular chapter or section contained in the Bentley Map Development and Product Customization Guide document.

1. Introduction > 4. How To Use This Guide? - To familiarize yourself with the navigation capabilities of this guide, click on a specific section title on this page, then click the chapter title. Similarly, using the "Table Of Contents"... (more)

3. XFM Introduction > 7. Bentley Geospatial Administrator - The Bentley Geospatial Administrator application can be used to create a new geospatial schema or edit an existing one. It can also be used to import an existing MicroStation GeoGraphics legacy proje... (more)

3. XFM Introduction > 6. XFM Projects - Use of an XFM project is the recommended practice when the application involves building or maintaining an intelligent map or infrastructure model. The XFM project and associated metadata ensures the... (more)

4. XFM Schema Introduction > 12. Property Based Annotation - The PBA expressions are similar in many ways to the MicroStation displayable attribute functionality but offer significantly more flexibility

Document Searches

The following links can be used to perform searches of commonly referenced content in the Bentley Map Development and Product Customization Guide document.

Archives - A listing of arc
within this document.

Chapters - A listing of this
document content includi
Contents" section that inc
Chapter entries.

Chapters and Sections -
guide with all document c
"Table Of Contents" secti
both Chapter and Section

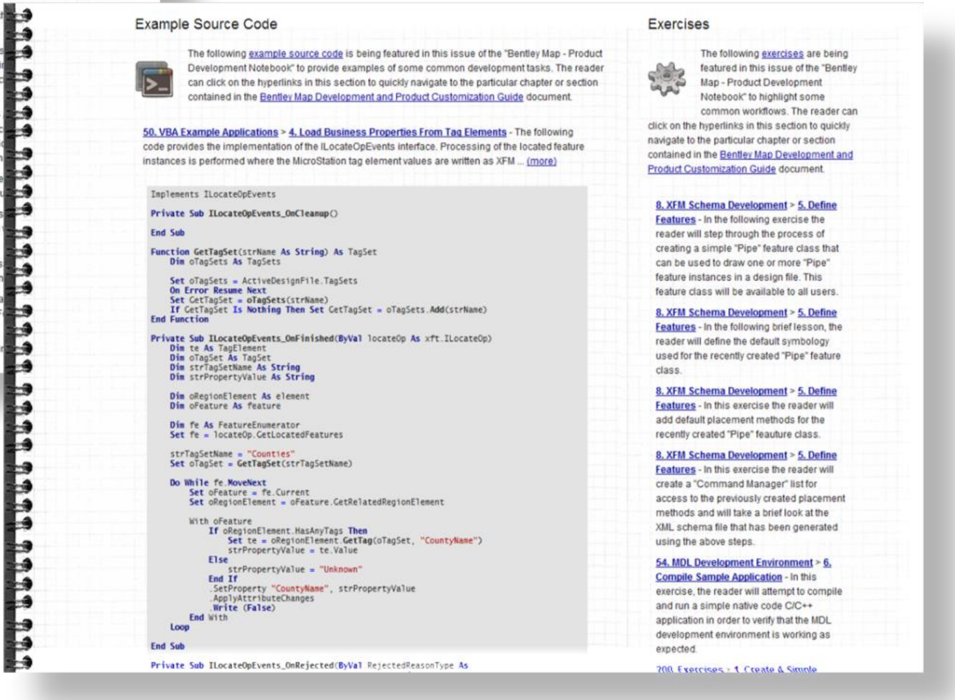
Code - A listing of sample
contained within this docu

Code - MDL - A listing of s
code snippets contained
document.

Code - VBA - A listing of s
snippets contained within

FeatureEnumerator - Exa
code making use of the F
object.

Links - A listing of hyperli
within this document.



Example Source Code

The following example source code is being featured in this issue of the "Bentley Map - Product Development Notebook" to provide examples of some common development tasks. The reader can click on the hyperlinks in this section to quickly navigate to the particular chapter or section contained in the Bentley Map Development and Product Customization Guide document.

50. VBA Example Applications > 4. Load Business Properties From Tag Elements - The following code provides the implementation of the ILocateOpEvents interface. Processing of the located feature instances is performed where the MicroStation tag element values are written as XFM ... (more)

```
Implements ILocateOpEvents

Private Sub ILocateOpEvents_OnCleanup()

End Sub

Function GetTagSet(strName As String) As TagSet
    Dim oTagSets As TagSets

    Set oTagSets = ActiveDesignFile.TagSets
    On Error Resume Next
    Set GetTagSet = oTagSets(strName)
    If GetTagSet Is Nothing Then Set GetTagSet = oTagSets.Add(strName)
End Function

Private Sub ILocateOpEvents_OnFinished(ByVal locateOp As xft.ILocateOp)
    Dim te As TagElement
    Dim oTagSet As TagSet
    Dim strTagSetName As String
    Dim strPropertyValue As String

    Dim oRegionElement As element
    Dim oFeature As feature

    Dim fe As FeatureEnumerator
    Set fe = locateOp.GetLocatedFeatures

    strTagSetName = "Counties"
    Set oTagSet = GetTagSet(strTagSetName)

    Do While fe.MoveNext
        Set oFeature = fe.Current
        Set oRegionElement = oFeature.GetRelatedRegionElement

        With oFeature
            If oRegionElement.HasAnyTags Then
                Set te = oRegionElement.GetTag(oTagSet, "CountyName")
                strPropertyValue = te.Value
            Else
                strPropertyValue = "Unknown"
            End If
            .SetProperty "CountyName", strPropertyValue
            .ApplyAttributeChanges
            .Write (False)
        End With
    Loop

End Sub

Private Sub ILocateOpEvents_OnRejected(ByVal RejectedReasonType As
```

Exercises

The following exercises are being featured in this issue of the "Bentley Map - Product Development Notebook" to highlight some common workflows. The reader can click on the hyperlinks in this section to quickly navigate to the particular chapter or section contained in the Bentley Map Development and Product Customization Guide document.

8. XFM Schema Development > 5. Define Features - In the following exercise the reader will step through the process of creating a simple "Pipe" feature class that can be used to draw one or more "Pipe" feature instances in a design file. This feature class will be available to all users.

8. XFM Schema Development > 5. Define Features - In the following brief lesson, the reader will define the default symbology used for the recently created "Pipe" feature class.

8. XFM Schema Development > 5. Define Features - In this exercise the reader will add default placement methods for the recently created "Pipe" feauture class.

8. XFM Schema Development > 5. Define Features - In this exercise the reader will create a "Command Manager" list for access to the previously created placement methods and will take a brief look at the XML schema file that has been generated using the above steps.

54. MDL Development Environment > 6. Compile Sample Application - In this exercise, the reader will attempt to compile and run a simple native code C/C++ application in order to verify that the MDL development environment is working as expected.

200. Exercises > 1. Create A Simple

Be Together - Development Workshop

MAY 23-26, 2011 | PHILADELPHIA, PA USA

Be Together

THE BENTLEY USER CONFERENCE

MyOracleExtensions.cs Start Page

My.Oracle.Extensions.OracleSpatialExtensions

```
        // Method used to perfor
        public bool connectSchem
        {
            bool opened = false;

            this.numChangesNew =
            this.numChangesModifi
            this.numChangesDeleted

            // Enable the option w
            XFMSTORAGE.ECObjectToX

            if (GDI.GeoDataIntercha
                return true;

            // This code opens the
            if (GDI.GDIExplorerAddIn
            {
                if (GDI.GDIExplorerA
                {
                    BGE.ExplorerTree
                    if (form == null)
                        return false;
                }

                if (GDI.GDIExplorerAdd
                    opened = (GDI.GDIE
            }
            return opened;
        }

        #endregion
```

9. Using the "**Build > Rebuild Solution**" command, results in the "**Output**" window, searching for any p

10. Press "**F5**" key to start a Bentley Map session with a workspace.

11. Select and open the existing **work.dgn** design file.

12. Select the "**Utilities > Macro > Project Manager**" **Manager**" dialog.

13. Select the "**Load Project**" icon on the button bar and and select and open the "**oracle1.mvba**" file.

14. In the "**VBA Project Manager**" dialog, select the "**ora** "**Microsoft Visual Basic Editor**" icon in the button ba **Editor**" integrated development environment.

Bentley

© 2011 Bentley Systems, Incorporated.

28

May 23-26, 2011 | Philadelphia, PA, USA

Be Together

THE BENTLEY USER CONFERENCE

SHARPEN YOUR EDGE

GE3WK1 – Bentley Map Development

Jeff Bielefeld, Bentley

1

Bentley

MAY 23-26, 2011 | PHILADELPHIA, PA USA

gether

ENTLEY USER CONFERENCE

SHARPEN YOUR EDGE

e: Developing Native C/C++ Applications for Custom r Functionalities

TIVE:

student will compile and review in the Visual Studio 2005 debugger some native C code the Map Manager API functionality of Bentley Map. The provided sample application to use the Map Manager API functions to create new map models, create map layers, ology, generate buffers and perform spatial overlay operations.

TING A NEW MAP MODEL (5 MINUTES)

will guide the student through the process of creating a new map model using the API.

w "**Microsoft Visual Studio 2005**" session and open the **_mapmaker1.sln**" solution file located in the "**c:\source\devenv**" folder.

to begin a new debugger session.

Station Manager, create and then open a new "**map1.dgn**" design file.

wly created "**map1.dgn**" design file has been opened, press the "**Schema** nd "**Query With Criteria**" buttons to query some Oracle Spatial feature instances the remaining exercises of this section, resulting in data similar to that shown in image.

Bentley

© 2011 Bentley Systems, Incorporated.

34

Bentley Sustaining Infrastructure

Additional Resources

- Be Communities
  - Geospatial Desktop Forum
    - http://communities.bentley.com/products/geospatial/desktop/f/5924.aspx
  - Bentley Developer Network Group
    - http://communities.bentley.com/programs/bentley_developer_network/default.aspx

- Direct E-Mail
  - bdn@bentley.com
  - jeff.bielefeld@bentley.com

Thank You