

(Paper # 1036, 2006 ESRI International User Conference, San Diego, CA)

Accessing Maryland Cadastral Data using ESRI Mapping Components

Kuang-Yao Lee
410-767-1218
klee@mdp.state.md.us

Mike Lettre
410-767-4460
mlettre@mdp.state.md.us

John Coleman
410-767-4614
jcoleman@mdp.state.md.us

Bob Dadd
410-767-4455
bdadd@mdp.state.md.us

Maryland Department of Planning
301 W. Preston Street, Suite 702
Baltimore, MD 21201
USA

Abstract

Since 1996 the Maryland Department of Planning has electronically maintained a statewide set of georeferenced property maps and parcel databases for Maryland's jurisdictions. Now completing the 10th Edition Year, the property maps and parcel databases along with a variety of related map layers are annually updated and released on CD-ROM for each jurisdiction. The property maps appear on the CD as georeferenced raster images in TIFF format and the parcel databases point shapefiles referenced by x,y to their location on the property map images. To increase the data accessibility, we developed a Windows-based desktop application that specifically displays the property map images and supports spatial queries of the parcel databases. This application is developed and maintained using Microsoft Visual Basic and ESRI MapObjects LT, which provides viewing, selecting, and printing capacities for spatially referenced property assessment information, market sales and transfer records, and detailed Computer-Aided Mass Appraisal data (CAMA). In addition, it efficiently displays seamless tax images with a transparent background, which consequently facilitates overlaying effects of map layers. This paper focuses on the approaches adopted to utilize Windows API and the downloadable ESRI ArcScripts to achieve desired functionality and improve performance.

Keywords: ArcScripts, MapObjects LT, Parcel, Visual Basic, Transparent Tax Maps

Introduction

From persons to businesses to government leaders, geo-spatial data increasingly provides invaluable information serving as a foundation upon which sound decisions are made in a more timely and effective manner. Geo-spatial data rely on computing resources on advanced computers for data display, manipulation, and storage. Along with computer enhancements, the software for display and storage of geospatial data has rapidly evolved in recent years.

Due to the wide range of uses and the large number of users, GIS technology has been significantly employed to maintain, update, and manipulate parcel property and land records information. Recent examples and various designs of GIS (including Desktop and Web-based) applications on property parcel information can be found in published literature and conference proceedings (see Razem et al. 2004; Tudor et al. 2004; Gisladottir 2005; Johnson 2005; Morgan 2005; Swinney and Kennedy 2005).

In 1993, the State of Maryland Department of Planning (MDP) initiated the computerization of the State's property maps in conjunction with the Maryland State Department of Assessments and Taxation (SDAT). Since the 1950s SDAT had maintained property maps in the form of ink on mylar originals (over 2,800 generally at a scale of 1:7,200 with insert maps for towns at smaller scales for Maryland's 23 counties). The maps were converted to digital format using scanning technology. The Mylar originals were typically 60.9 cm x 91.4 cm (2 ft x 3 ft) in size and were scanned on a Contex FSS8000 scanner at 400 dots per inch (dpi). The FSS8000 scanner is an E-size monochrome scanner capable of 800 dpi. The 400 dpi setting produced the best results with an acceptable file size and reasonable clarity of printed output.

Since 1993 the maps have been maintained as a hybrid of the original raster scans with vector updates for all new subdivisions and property boundary changes. A number of areas have also been completely vectorized with parcel boundaries aligned to digital ortho imagery. Over the last 12 years (over 500,000 parcels) have been vectorized. The current software platform is Autodesk Map 3D 2006 with Raster Design 2006. The use of Raster Design allows the combined raster/vector image files to be edited in either file format -- the vector portion residing in DWG file format and the raster image portion in a TIFF Group 4 compressed format.

To make the computerized property maps intelligent, each geographically referenced property map along with State Highway Administration's 1:24,000 scale digital roads files are used to enter x,y parcel reference points. The parcel reference points are within the lot lines of the parcel as it appears on the property maps such that it also falls on the correct side of the digital roads file to the maximum extent possible. The referenced points are linked via a unique identifier to the corresponding parcel database record for the property. To date, this has been achieved for nearly 99% of the State's 2.2 million parcel database records.

The above procedure has proven to be a comparatively cost effective way to computerize the maintenance of the State's property maps and to create an intelligent, geographic link to the existing automated parcel database records. Given the uncertain quality and map accuracy of the property maps, the parcel x, y reference points along a statewide 1:24,000 digital road file proved to be an achievable and doable scale.

Since 1996, MDP has been exploring more cost effective ways to make the computerized property map and parcel data accessible to public, not-for-profit and private sector users as well as general public. The goal is to provide the electronic property maps and geospatial parcel data sets to users in a readily accessible manner using affordable desktop viewing software and standard personal computers. The desktop software package - ArcView 3.x with customized project file (.aprs), developed by Environmental Systems Research Institute, Inc. (ESRI), proved to be the best initial choice (in 1996) for initially distributing these geospatial datasets on CD-ROM. Among its benefits were relative ease of use, ability to handle a full range and variety of file formats for the collection of data layers to be distributed and general affordability. Since 2002 the data sets on CD-ROM for each Maryland county and Baltimore City have also included a customized ArcGIS 9.x mxd for use in accessing, querying and viewing the parcel data, property maps and related data sets.

Each year's new edition includes updated property maps and completely refreshed parcel database records for each of Maryland's 24 jurisdictions (2-3 counties are released each month during an edition year beginning in July and ending in June). The parcel data are derived from snapshots of the State's assessment database just before final production of a jurisdiction's CD-ROM. To keep the CD-ROM product up-to-date over the course of the year, a sales transaction file containing information for the parcels that have been sold over the prior 12 months is generated each month. This file is populated with the x,y reference points that link the parcels to their location on the property maps when viewed using the ArcView or ArcView GIS software. The sales files are broken down by jurisdiction and are loaded onto a server in the Department of Planning's website. Through the Internet connections, users can download the latest jurisdiction sales file each month and add the database to their ArcView project files for analysis. In addition the customized .aprs and .mxd allow the user to select a parcel and see it displayed on a map using the CD-ROM data contents and then go directly to a live link for that parcel showing the latest SDAT record information on their real-time property look-up website.

Prior to 2003, viewing the MDP property maps and parcel data relied almost exclusively on ESRI desktop Mapping Products - either ArcView GIS 3.x or ArcGIS 9.x. In 2002 MDP began development of a new software tool for accessing these data sets, a simple, lightweight Windows desktop GIS viewer. The purpose in developing the GIS viewer was to increase accessibility, simplify operations, reduce software complexity and cost, and shorten user training and the learning curve. The GIS viewer, using recent mapping technologies, is a simple and easy-to-use Desktop Application for accessing Maryland Statewide property parcel information. In 2002 MDP developed a prototype of the GIS viewer (named WebCompanion and subsequently renamed FINDER upon the addition of expanded functionality beginning in 2003). The viewer software has evolved over the last several years and is now in its 4th edition with the recent release of FINDER 2006. This paper focuses on the approaches used during the development and evolution of this software and the solutions for the technical problems encountered.

Methods

Software Development

To develop the framework for the FINDER application, a modified Waterfall model (see McConnell, 1996) was used. Basically, it consists of the following four steps from development to deployment:

1. Requirements Engineering

The requirements engineering at the development stage for FINDER analyzed the problem domain and defined the requirements (Kosters et al., 1995). Learning from years of experience with users accessing and viewing our datasets with ArcView 3.x and ArcGIS 9., key user needs were identified that needed to be addressed. Specifically these needs were ease of use in viewing, selecting, and printing property maps and associated parcel and sales data. Secondly, the file format for this application and its datasets needed to be easily accessible and transportable for data sharing. Finally, an understandable and widely used computing language was needed that would be transparent and easy to maintain. i.e. these three needs drove the requirements analysis.

2. Development Environment

The FINDER software design has to work with annually updated geospatial datasets prepared by MDP for each Maryland jurisdiction (23 counties Baltimore City, see Table 1). Each jurisdiction dataset is released on a CD-ROM with the size of datasets ranging from 150 to 600 Mb depending on the jurisdiction. The FINDER software also has to work with a variety of PC hardware and versions of OS systems. FINDER is specifically compiled under the Windows NT environment and is expected to be compatible with NT systems and above.

3. Implementation

FINDER is designed to have a main Window that contains a map canvas (a display object of MapObjects LT) to display shapefile layers. The majority of the Global variables used in FINDER are declared in the main Windows (Form). The coding statements are developed primarily based on MapObjects LT documents published by ESRI (1999; 2000a & 2000b) and discussions from ESRI's Downloads, Support, and Forums (<http://www.esri.com>). Microsoft Package and Development Wizard from Visual Studio 6 is used for compilation (see <http://support.microsoft.com/>). This implementation stage includes critical design review.

4. Testing & Deployment

Three stages of testing are used in preparing a new release of the FINDER software: 1) a raw test by developers; 2) a functionality test for specific application features by a small group of internal staff; 3) a broader test by external beta users. The purpose of this phased testing is to ensure the application meets and performs as expected by users (Kosters et al. 1995)

Accessing County Datasets

To provide an easy spatial referenced view for each jurisdiction, FINDER's map view as it appears when initially opened automatically loads 5 essential Shapefiles and one image. These shapefile are Property Parcel Point, Sale Data, Tax Map Grid Index, ADC Page Index, County Boundary, and the scanned Maryland State Highway Administration road map. This provides users with a full county view with State Highway roads and features as the background for spatial reference (see Figure 1). A dropdown menu located at the upper right hand of the main window of FINDER allows user to make these layers visible (show) or invisible (hide). Users are able to add and display additional layers (shapefiles) in the map view using the additional layers functions (not discussed further since it is beyond the limited scope of this paper).

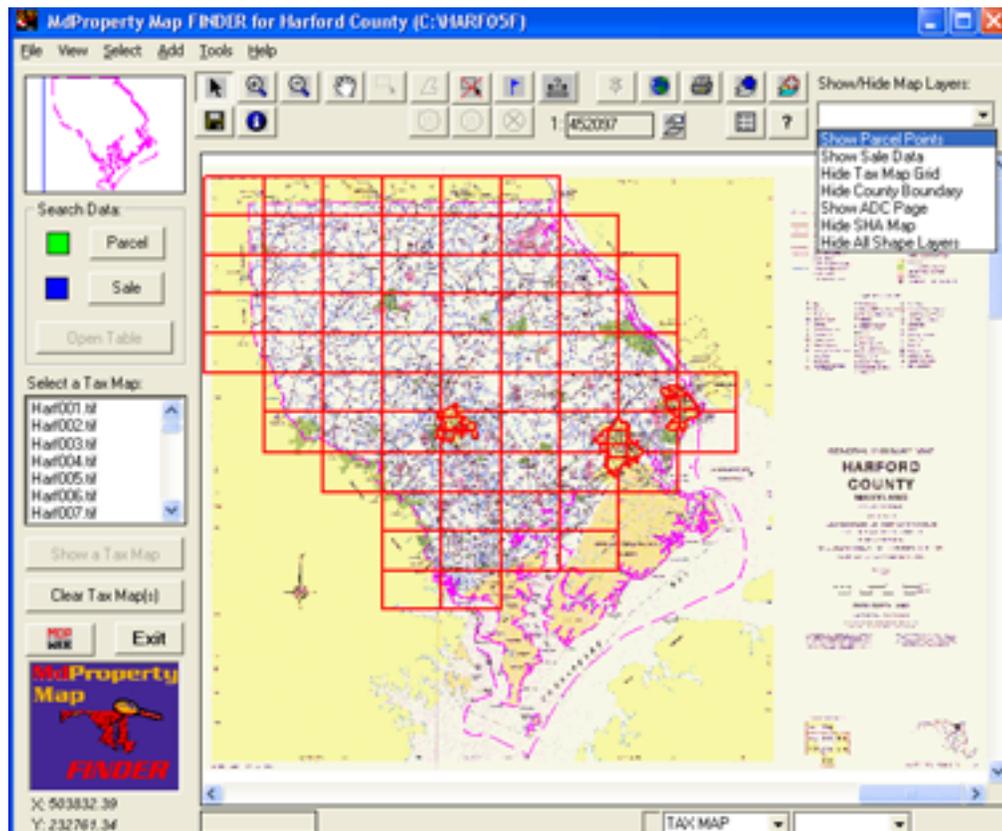


Figure 1. The main canvas window provided from MapObject LT in the FINDER application. The upper right dropdown menu allows users to turn layers on or off in the map view. The Tax Map Grid, County Boundary, and Scanned SHA Color Map layers are turned on; the Parcel, Sale and ADC Page layers are turned off in the opening map view.

Issues Encountered

Several performance issues were encountered during the initial development and testing stages: 1) slow search performance on the parcel point layer (Shapefile .dbf) due to the large numbers of records, over 300,000 for the largest jurisdiction and the various data transfer rates (megabytes per second) on CD-ROM drives. 2) lack of transparency on Tax Map (.TIF) images, and 3) inability to create Shapefiles that can

append additional parcel attribute data (Computer Assisted Mass Appraisal data) when exporting.

Search

Based on the 2005 Edition jurisdiction datasets, the numbers of parcel records range from 13,009 for Kent County to 318,337 for Montgomery County. The search performance degrades for larger counties and particularly for data layers that are accessed directly from CD-ROMs drives. Table 1 shows a preliminary test on search performance of the FINDER application on two different configurations of personal computers. As shown in Table 1, search time increases when using the lower end personal computer, in particular for larger jurisdictions (i.e. Anne Arundel County, Baltimore City, Baltimore County, Montgomery County, and Prince George's County).

Table 1. A simple, preliminary study of search performance on different personal computers was tested by directly accessing data layers from CD-ROM drives. A SQL statement, OWNNNAME = %smith%, was entered in the MapObjects LT search engine from the compiled version of FINDER. Time measurements ended when the record set returns the number of selected records from the attribute table of the Shapefile. The bracket of % in the SQL statement represents a wildcard search. OWNNNAME is one of the database fields in the parcel point Shapefile. The number of parcels in the attribute table for each Maryland jurisdiction as of the latest edition is shown (as of May 2006).

County Name	# OF PARCELS	Time (sec) needed to get results	
		2.8GHz, 1 RAM CPU: Intel P4	1.7GHz, 256 RAM CPU: Intel P4
Allegany	40876	8	11
Anne Arundel	196352	21	50
Baltimore City	234157	32	77
Baltimore	289747	42	88
Calvert	40609	7	12
Caroline	15995	4	6
Carroll	64524	9	23
Cecil	44920	7	12
Charles	56825	8	21
Dorchester	20905	3	9
Frederick	88116	12	53
Garret	27767	5	9
Harford	91011	15	31
Howard	94297	15	45
Kent	13009	3	4
Montgomery	318337	40	93
Prince George's	272403	35	90
Queen Anne's	23950	5	8
Somerset	16398	4	10
Saint Mary's	42943	7	15
Talbot	19962	3	6
Washington	57200	10	21
Wicomico	43924	7	13
Worcester	61113	11	19

From a user's point of view, a response time of 20 seconds to get search results is reasonably acceptable. The time needed is only in response to approximately 50,000 parcels in the database (see Figure 2). Searches above 50,000 parcels resulted in a significant increase in the response time, in particular for lower-end personal computers (~3 fold increase in time). This may lead to a potential issue for increasing parcel numbers as a result of more subdivisions and may represent a search performance issue for some users.

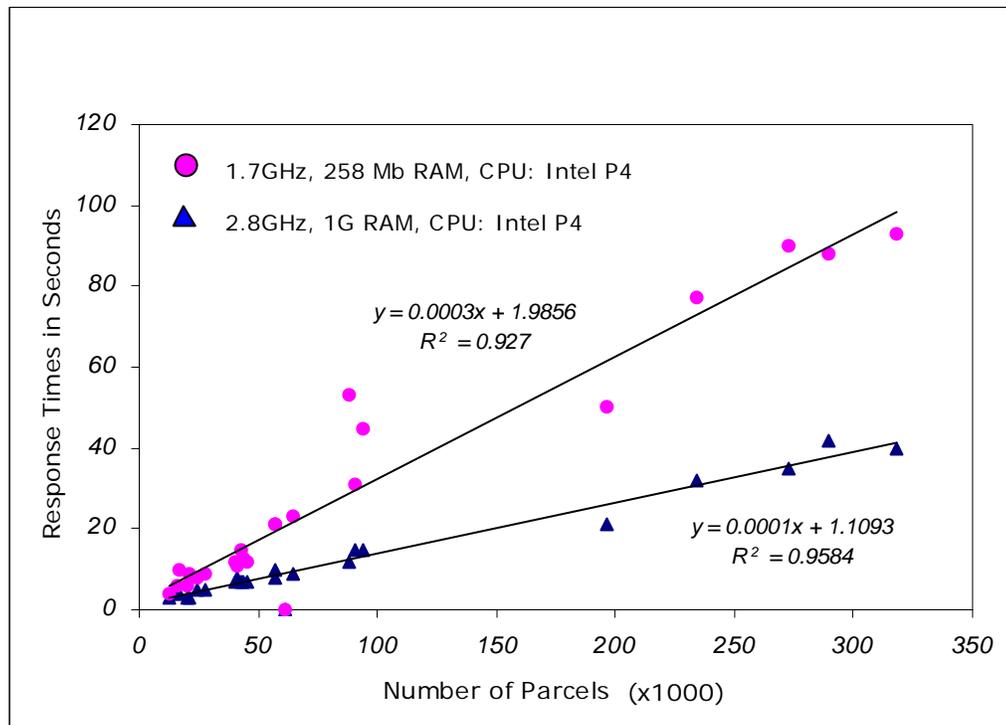


Figure 2. Linear regressions between response times and the numbers of parcels from two different personal computers. Plots were based on the results of Table 1.

Image

Each property map is provided in a jurisdiction's dataset in the format of a TIFF image that contains pixel values ranging from 0 to 1 in which 0 represents White background and 1 represents Black boundaries on the property map images. Since the map images often have a border with a 0 or "white" color that extends out beyond the "black" areas, one map image may cover up some of "black" lines showing property boundaries when two or more adjacent map images are displayed at the same time (see Figure 3). This also results in hindering visual effects on overlaying with other layers. For example, the Land Use Land Cover layer is not invisible or hidden below the tax map images (see Figure 3). The seriousness of this problem is variable depending on differences in the width of the borders on the tax map image.



Figure 3. Two tax map images align side by side in non-transparent conditions, which results in the gaps between two images where property boundary lines are broken. In the example above the gap is small but in some instances the gap is considerably larger.

Export

Due to the limitations of MapObjects LT, exporting selected records is limited to Text File (ASCII) or Dbase format Table (DBF) while using Microsoft DAO (Data Access Objects, Microsoft 1997) within the Visual Basic environment. For the application to meet user needs, each residential parcel record needs to have the option to attach CAMA (Computer Assisted Mass Appraisal) information from the CAMA layer. This is required to perform a spatial query using MapObjects LT while exporting is in process.

Solutions

Limit the Search Extent

Search performance efficiency is improved when the search limits are set based on the current map view extent. As shown in Figure 4, the search expression of "SearchShape" is coded to limit the selection of parcel points within the current map extent (rectangle). Thus, it depends upon the map scale to determine the number of records to be searched, which eliminates unwanted locations of parcels points from the search. This partial Visual Basic procedure in Figure 4 also demonstrates the search within the selection set (i.e. subset). EXP presents the search query in SQL strings from the input dialog box.

```

Dim rectangle As MapObjectsLT2.Rectangle
Set rectangle = Map1.Extent

'..Search within the extent
Dim subset As MapObjectsLT2.Recordset
If OptSearch(0).Value = True Then
    If pSearchWithin = True Then
        Set subset = g_pRS
        Set g_pRS = Nothing
        Set g_pRS = frmMdPVMMain.Map1.Layers("PROPDATA")._
            SearchShape(subset, moContaining, EXP)
    Else
        Set g_pRS = Nothing
        Set g_pRS = frmMdPVMMain.Map1.Layers("PROPDATA")._
            SearchShape(rectangle, moContaining, EXP)
    End If
End if
....

```

Figure 4. The Visual Basic code to limit the search of parcel records to the extent of the current Map display. Map 1 shows the Map view with the full extent. PROPDATA is the parcel point layers in the format of a Shapefile when shown (made visible) in the Map.

Transparent Images

In order to make the white portion of the property map images transparent, one of Windows Application Interfaces (APIs) was adopted - Bit blit, exposed by Windows GDI32.DLL (see Appleman 1996; Appleman & Grimes 1999; Appleman et al. 1999; Roman 2000; Bai 2003). This is one of the core sets of application programming interfaces in the Microsoft Windows Operation System. To access this Graphics Device Interface (GDI) DLL, the declaration statements must be placed on the top of Visual Basic form or module (Figure 5). This operation enables FINDER to copy, combine, or merge rectangles of bits between main and display memory on a bit-mapped device.

```

Private Declare Function CreateCompatibleDC& Lib "GDI32" _
    (ByVal hDC As Long)

Private Declare Function CreateCompatibleBitmap& Lib "GDI32" _
    (ByVal hDC As Long, ByVal nWidth As Long, ByVal nHeight As
Long)

Private Declare Function BitBlt& Lib "GDI32"
    (ByVal hDestDC As Long, ByVal X As Long, _
    ByVal Y As Long, ByVal nWidth As Long, _
    ByVal nHeight As Long, ByVal hSrcDC As Long, _
    ByVal xSrc As Long, ByVal ySrc As Long, _
    ByVal dwRop As Long)

Private Declare Function SelectObject& Lib "GDI32" _
    (ByVal hDC As Long, ByVal hObject As Long)

Private Declare Function SaveDC& Lib "GDI32" _
    (ByVal hDC As Long)

Private Declare Function RestoreDC& Lib "GDI32" _
    (ByVal hDC As Long, ByVal nSavedDC As Long)

Private Declare Function DeleteDC& Lib "GDI32" _
    (ByVal hDC As Long)

Private Declare Function DeleteObject& Lib "GDI32" _
    (ByVal hObject As Long)

Private Declare Function GetClipBox& Lib "GDI32" _
    (ByVal hDC As Long, lpRect As rect)

```

Figure 5. The use of GDI32.DLL needs declaration of functions on the top of Declaration in Visual Basic Form or Module during the development phase. These standard statements are easily found from various resources such as Internet websites and programming reference books (Roman 2000; Appleman & Grimes 1999; Appleman et al. 1999, Bai 2003).

FINDER executes the Bit Blit statements within the procedures of Map refresh (Figure 6) in which BitBlt copies and stores the images into memory before the map layers redraw (BeforeLayerDraw) and pastes back to Map (in a Form) from memory after the layers redraw (AfterLayerDraw).

```
Private Sub Map1_BeforeLayerDraw(ByVal Index As Integer, ByVal  
hDC As StdOle.OLE_HANDLE)
```

```
    xIDCSaved = SaveDC(hDC)  
    xdl& = GetClipBox(hDC, rectDC)  
    IWidthDC = rectDC.Right - rectDC.Left  
    IHeightDC = rectDC.Bottom - rectDC.Top  
    xdcNew = CreateCompatibleDC(hDC)  
    xbmpNew = CreateCompatibleBitmap(hDC, IWidthDC, IHeightDC)  
    SelectObject xdcNew, xbmpNew  
    xdl& = BitBlt(dcNew, 0, 0, IWidthDC, IHeightDC, _  
        hDC, 0, 0, &HCC0020)  
    RestoreDC hDC, xIDCSaved  
    DeleteObject xbmpNew
```

```
End sub
```

```
Private Sub Map1_AfterLayerDraw(ByVal Index As Integer, ByVal  
Canceled As Boolean, ByVal hDC As StdOle.OLE_HANDLE)
```

```
    Dim xIDCSaved As Long  
    Dim xdl&  
  
    If Map1.Layers(Index).LayerType = 1 And _  
        UCase(imgPath) = UCase(myPath) Then  
  
        xIDCSaved = SaveDC(hDC)  
        xdl& = BitBlt(hDC, 0, 0, IWidthDC, IHeightDC, _  
            dcNew, 0, 0, &H8800C6)  
        RestoreDC hDC, xIDCSaved  
        DeleteDC dcNew  
    End if
```

```
End sub
```

Figure 6. The Bit Blit coded in the event procedures of the Map – BeforeLayerDraw (upper panel) and AfterLayerDraw (lower panel) of the Map

Export using ESRI ArcScripts

The ESRI ArcScripts (<http://arcscripts.esri.com/>) site provides useful resources for GIS application developers. Downloadable VB class modules (ShapeIO, 1998 version, created by Kenneth R. McVay) were adopted for the FINDER application to facilitate the export of selected data. However, this posted ShapeIO is only to read and write two shapefile sublayers (i.e. .SHP and .SHX). In order to meet the minimum requirement of a Shapefile as described in ESRI *Shapefile Technical Description* (ESRI 1998), Microsoft DAO was used to create the corresponding .DBF for the resulted Shapefile. Since the spatial location of a parcel point stores the real-world coordinates, FINDER deciphers the x,y coordinates from the [Shape] field of selected records and writes the coordinates to the new .DBF for the resultant Shapefile. The resultant Shapefile; therefore, contains 3 fundamental files (i.e., .SHP, .SHX, and .DBF) that comprise of the simplest form of a Shapefile (ESRI 1998).

Spatial query to attach CAMA Data

The residential CAMA data layer is a separate Shapefile that is provided and stored under the same data folder where the Parcel and Sales data layers are located. The CAMA attributes are optionally attached and exported along with property parcel or sale points while exported out to a Text file (.TXT), Dbase table (.DBF), or Shapefile. To attach the CAMA attributes, the spatial query using MapObjects LT map component is employed when performing this option. First, the real world x,y coordinates of each of the selected parcel points from the Map View are recorded into memory where a virtual rectangle (side = 50 meters) forms with the x,y coordinate of the selected parcel point as the center. After loaded into the map, this rectangle is then used to perform a spatial search query on the CAMA data layer with a match of account ID with parcel data points since both the parcel point and CAMA layers have a "common and unique" field - an unique property account ("ACCTID" and "ACCTCAMA", respectively). The match on the account ID is required since stacked parcel points (multiple parcel points each with the same x,y location but a unique account, e.g. condominiums) can occur for some locations.

The partial Visual Basic code in Figure 7 demonstrates the processing steps to attach CAMA attribute data to selected parcel points when both are to be exported out in response to a FINDER user's request. This looping procedure executes until the end of selected records. Here is the summary of the steps:

1. The attributes of the selected parcel records show in a ListView table (a Visual Basic Object) where x,y coordinates are stored.
2. A virtual rectangle (i.e. square) forms based on the x,y with 50 meters on each side.
3. Map zooms to the selected point along with the center of the virtual rectangle.
4. In the SQL expression, a unique account ID from the parcel is parsed to match account ID (ACCTID) to the account field (ACCTCAMA) in the CAMA attribute table.
5. The application executes a spatial query with a match of unique IDs.
6. The selected attributes from CAMA are exported along with the parcel information.
7. The loop cycle executes for each selected record in the ListView.

```

...
Dim pX As Long
pX = Val(ListView1.SelectedItem.ListSubItems(18).text)

Dim pY As Long
pY = Val(ListView1.SelectedItem.ListSubItems(19).text)

..define a rectangle
Dim ptBounds As MapObjectsLT2.Rectangle
Set ptBounds = New Rectangle
ptBounds.Left = pX - 50
ptBounds.Right = pX + 50
ptBounds.Top = pY + 50
ptBounds.Bottom = pY - 50

'..preserve the current extent
Dim myExtent As MapObjectsLT2.Rectangle
Set myExtent = frmMdPVMMain.Map1.Extent

'..zoom to the defined extent
frmMdPVMMain.Map1.Extent = ptBounds
frmMdPVMMain.Map1.CenterAt pX, pY

Dim mdpvexp As String
mdpvexp = "ACCTCAMA = '" & _
          (ListView1.SelectedItem.text) & """"

Dim CAMARecSet As New MapObjectsLT2.Recordset
Set CAMARecSet = frmMdPVMMain.Map1.Layers("CAMADATA"). _
                SearchShape(ptBounds, moContaining, _
                mdpvexp)
CAMARecSet.MoveFirst
...

```

Figure 7. The partial Visual Basic code in the procedure that performs spatial query on the CAMA data layer.

Results

The three major performance issues encountered were successfully resolved using available resources from Internet search and references books (Roman 2000; Appleman & Grimes 1999; Appleman et al. 1999, Bai 2003). First, adjusting the map extent before performing a search on the parcel or sale datasets substantially shortens time needed for database searches. Search within selected set also allows the search to refine the initial selection. Second, the transparent property maps resolved overlaying problems when multiple, adjacent property map images are displayed or when other layers are displayed that are in a location under the tax maps. As shown in Figure 8, there are 2 adjacent tax maps (.TIFF) aligned seamlessly after FINDER executes Bit Blit API function. The transparent conditions enable Map to display land use land cover (shaded – Forest and light shaded – Agriculture) underlying the tax map images, resulting in better visual effects on display.



Figure 8. Two tax map images align side by side with transparent conditions, which result in one complete image visually.

Third, the ability to export selected records to shapefiles was implemented using ESRI downloadable ESRI scripts. This enables additional attributes to be appended using a spatial query statement of MapObjects LT when selected records are being exported.

FINDER Product Life Cycle (FPLC)

The Support Policy is developed to help communicate to users with the technical support resources available (PSC 2005) during FINDER's life span. It also provides advanced notification of planned changes to available support options. The FINDER Product Life Cycle Support Policy has 4 phases during each one-year life cycle (July to June) as summarized below:

1. Availability for the public (June)
 - a. Official announcement of release via website or mails
 - b. This is the Beta version - Phase II
 - c. Availability of and notification of training classes
2. Extended Support (3 months from the release day)
 - a. Phone support
 - b. E-mail
 - c. Training for core functions
 - d. Fixes
3. Mature Support (9 months from the end of extend support)
 - a. Phone support
 - b. E-mail
 - c. Training for core functions
 - d. No fixes

4. Retired (after the first release of next generation/edition)
 - a. No longer available to users
 - b. Phone support if still used
 - c. Training for core functions only
 - d. No fixes

For FPLC support status, currently the FINDER 2005 software is approaching the retired stage since the first edition of the FINDER 2006 software has been announced. All the retired editions are no longer supported although they are still available (see Table 2).

Table 2. The Support status of FPLC as of June 2006

Version	Name	Availability	Extend Support	Mature Support	Retired
2002	Web Companion				June 2003 (Retired version)
2003	FINDER 2003				June 2004 (Retired version)
2004	FINDER 2004				June 2005 (Retired version)
2005	FINDER 2005	X (July, 2005)		X	
2006	FINDER 2006	X (June, 2006)			

Conclusions

FINDER provides the fundamental GIS features that allow users to zoom in, zoom out, pan, select, query, print, and report property parcel information referenced to property maps images. Other features include the ability to search using SQL expressions, to link to web URLs, and to generate tables from Property or Sales databases.

Although, FINDER does not possess advanced GIS capabilities (i.e., data manipulation, construction of polygon topology, advanced geoprocessing, and coordinate projections, and only works with Shapefiles), the original capabilities of MapObjects LT have been enhanced and expanded by using available computing resources. These enhancements have made this application an attractive alternative to users wishing to access our datasets but lacking the expertise, interest, need or resources to use true desktop GIS software (ArcView 3.x or ArcGIS9.x).

Finally, it is noted that a reliable and efficient software product to support "map viewing" of MDP's property map, parcel database and related layers requires

thoroughly evaluating many factors that contribute to its acceptance and successful use by users. This paper merely provides technical background information on the major issues encountered during the FINDER software development process. It does not cover the many other components critical to the acceptance and use of this product including, data preparation, format conversion, layer editing, production, client tracking, communication & marketing, and support & training.

Acknowledgments

A special thanks to all and specifically to the following individuals whose dedication and hard work (including digitizing, data compilation, format conversion, programming, and document development) has made possible ten years of annual updates to the Property Map and Parcel point databases for Maryland's jurisdictions - Linda Nace, Jean Appleton, Carol Taylor, Gary Maragos, Al Blank, Rob Maxwell, Teresa Feetham, Mark Lebed, Lynn Bierly, Virginia Hawkins, and Celeste Ratti. We also extend our thanks to Kenneth R. McVay for posting his ShapeIO Visual Basic class module for free to download in the ESRI website.

Reference

Appleman D. (1996) Visual Basic Win32 API. Ziff-Davis Press, ZD Press, Emeryville, CA.

Appleman D., P. Baumgartner, P. Baumgartner (1999) WIN32 API Puzzle Book and Tutorial for Visual Basic Programmers. Apress L. P Publishing.

Appleman D., and G. Grimes (1999) Visual Basic Programmer's Guide to the Win32 API. Sams Publishing.

Bai Y (2003) Applications Interface Programming Using Multiple Languages: A Windows Programmer's Guide. Prentice Hall PTR, New Jersey.

ESRI (1998) ESRI Shapefile Technical Description.
<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. Redland, CA. 34 pp.

ESRI (1999) MapObjects Programmer's Reference. Environmental Systems Research Institute, Inc., Redlands, CA.

ESRI (2000a) MapObjects LT Object Diagram.
http://downloads.esri.com/support/whitepapers/mo_molt_dgm.pdf. Redland, CA.

ESRI (2000b) Getting Started with MapObjects LT 2 and Visual Basic 6.
http://downloads.esri.com/support/documentation/other_molt2gettingstarted.pdf
Redland, CA.

Kosters G., B.-U. Pagel, H.-W. Six (1995) Object-Oriented Requirements Engineering for GIS-Applications. *In* Proceeding of ACM-GIS International Workshop on Advances in GIS. Baltimore, MD.

Gisladottir G. (2005) Tracking and Managing Parcel Changes in a Geodatabase Format. *In* Proceeding of 2005 ESRI International User Conference. San Diego, CA.

Johnson R (2005) Minnesota MetroGIS Geospatial Data Collaborative Minneapolis – St. Paul Metropolitan Area (2002 – Enterprise System). Journal of the Urban and Regional Information Systems Association. Vol 17 No.2.

McConnell S (1996) Rapid Development: Taming Wild Software Schedules. Microsoft Press., Redmond, Washington.

Microsoft (1997) Microsoft Visual Basic 5.0 Programmer's Guide/ Reference Library. Microsoft Corporation. Redmond, Washington.

Morgan D. (2005) GIS-Driving Force of Land Records System in Beaufort County. *In* Proceeding of 2005 ESRI International User Conference. San Diego, CA.

PSC (2005) Progress Software Product Life Cycle Guide. Progress Software Corporation. (<http://www.progress.com/products/lifecycle/index.ssp>)

Razem F., D. Meierer, M Rose (2004) Developing an Integrated Real Estate Information System (IRIS) using ArcGIS. *In* Proceeding of 2004 ESRI International User Conference. San Diego, CA.

Roman S. (1999) Win32 API Programming With Visual Basic. O'Reilly Publishing.

Swinney J., J. Kennedy (2005) Geographic Information and Land Record Registration System for Wise County, Virginia. *In* Proceeding of 2005 ESRI International User Conference. San Diego, CA.

Tudor G., J. Devitt, F. Fischer (2004) Washington State Cadastre: Integrating Topology, Aquatic Ownership, and Land Records. *In* Proceeding of 2004 ESRI International User Conference. San Diego, CA.

About Authors:

Kuang-Yao Lee is the Manager of System Support and Product Development for the Maryland Department of Planning. He is responsible for GIS customization, spatial database maintenance, and desktop and online application development. He obtained a Ph.D. degree in Marine, Estuarine and Environmental Science from University of Maryland College Park in 2000.

Mike Lettre is the Director of Planning Data Services for the Maryland Department of Planning. His supervisory responsibilities include Property Mapping, Census, Redistricting, GIS, and Socio-Economic Projections. He is a past chair of the Maryland State Geographic Information Committee (MSGIC) and has served on two National Academy of Sciences Panels on the Decennial Census. He has a M.S. in public administration and a B.S. in business administration, from Carnegie-Mellon University.

John Coleman is the Director of Marketing & Outreach for the Maryland Department of Planning (MDP). He is responsible for coordinating all aspects of public outreach for the Department. He is principally involved in marketing of and subscriber support for MDP's electronic parcel-mapping GIS tools, MdProperty Map *FINDER* and

MdProperty View. He received his bachelor's degree in Business Administration from Frostburg State University in 1985 and has been with the Maryland Department of Planning since early 1999.

Bob Dadd is the Manager of the Property Mapping Section for the Maryland Department of Planning (MDP). He has supervised the completion of ten years of property map updates and the preparation of the maps, images, databases and GIS coverages that constitute the real property related products for the Department. He has a M.B.A. in Quantitative Methods and a B.S. in Statistics from the University of Maryland. Before joining the Department of Planning in 1990, he worked twenty years in several regional and local planning agencies in Southern Maryland and Fairfax County, Virginia.