

# Helpful Hints for AutoCAD Users in a Surveying Office

By Boris V. Gutkin, Ph.D.

## Introduction

Working and talking with a number of people from surveying firms in Canada and abroad, I have found that many people are purchasing software programs in an effort to meet deadlines, improve their bottom line and make their life easier by working with famous software programs.

We have all kinds of companies providing software specifically for surveying needs but we all find that they are not exactly meeting our needs. Surveying companies all have different ways to prepare a drawing and prepare their final product. Spending time and money for new modern software and training courses to teach our staff how to use the new products can be eliminated by learning a few simple techniques to create our own routines that will meet our specific needs.

How many times have you been under a tight deadline to deliver a finished drawing only to be informed by your draftsman that "the computer has crashed and I just lost a drawing" or "I need to change the scale and all of the text and blocks are in the wrong place - what should I do?"

The answer to these questions and many others is available to you with

some simple programming know-how. This paper will demonstrate an easy way with which it is possible to develop common functions to work with AutoCAD that will benefit the land surveyor.

## Visual Basic for Applications

One option to customize Windows based software is to use Visual Basic, one of the common languages supported by most Windows Applications. AutoDesk also provides the option to develop specific applications under AutoCAD and other products with Visual Basic.

There are a number of advantages to implementing Visual Basic for Applications (VBA) for AutoCAD. First and most importantly, Visual Basic courses are available in most surveying and engineering programs at colleges and universities. It is quite likely that some people in your office will already know how to write some simple procedures in VBA. Secondly, the programming language and development environment are easy to use and come installed with all AutoCAD Windows versions. This implies that we do not have to pay an additional fee for using VBA. The third advantage is

that there are numerous Visual Basic programmers around the world. We can easily get help in writing our own procedures from Internet user forums and special Internet sites. Some of them you will find in references for this topic. All of the procedures in this paper can be used in your AutoCAD as macros or by creating a VBA project.

## Saving Backup Files

We all know that when working with computer drafting, it is possible to have a situation where the program is terminated and we have no way to recover the corrupted file. It is possible to take precautions to minimize the potential harm. First, we have to make sure that in the File Safety Precautions tab of the AutoCAD's Options Dialog Box, we have the options "Automatic save" and "Create backup copy with each save" activated (checked). In addition, it is very important to insure the correct path for auto-saved files. However, this option will not work correctly if there was some kind of internal or other crash error. I have created a small procedure to save another copy of the backup file in a different location. This method (refer to **Figure 1**) is straightforward and all that is required is to create a macro under the Tools tab and make sure that it will execute each time you press the "save" button. Figure 1 uses the "C:\Autosave Files\" path to save files, but you can change it according to

```
Sub FileSafety()  
'Saving backup copy procedure *  
  Dim StrFileName As String  
  Dim fs As Object  
  On Error GoTo Err_Control  
  'Retrieving file name *  
  StrFileName = ThisDrawing.FullName  
  'Assigning backup copy's name *  
  StrFileName = Left(StrFileName, Len(StrFileName) - 4) + ".bak"  
  'Copying source file *  
  Set fs = CreateObject("Scripting.FileSystemObject")  
  fs.CopyFile StrFileName, "C:\Autosave Files\", True  
  Exit Sub  
Err_Control:  
  Select Case Err.Number  
  Case 53  
    Err.Clear  
    Exit Sub  
  Case Else  
    Exit Sub  
  End Select  
End Sub
```

**Figure 1**



your needs. Backup files with the same name will be overwritten and this option will minimize disk space use.

This routine does not fix a "crash error" but will keep your drawing in a secure location should a crash occur. It functions as an additional save control and, if the draftsman has saved the drawing, will keep the last saved version.

### Drawing scale changing

One of the most common problems that we face in each surveying office is scale changing. First, we have to decide which drawing objects will be scaled. We cannot scale linear objects because in most cases we are working at a scale of 1:1 within the drawing. Consequently, we should think about scaling only text entities and blocks. Unfortunately, AutoCAD does not have the option to scale groups of objects at the same time and use insertion points of each object as the base point for scaling.

Our procedure will process the entire drawing and select only blocks and text objects. As a second step, we will run a loop to scale each object and insure that it does not change position. This is very important when we are working with coordinates. Here we have two concerns. First, we should only select text objects and blocks. We can do that by defining rules for your selection set.

The code in **Figure 2** prompts users to select only text objects in the current layer to be included in a selection set. Using the examples in Figure 2 will allow you to create a project that will work like the "change scale" feature under one of the buttons in your AutoCAD's toolbar. The process of re-scaling will be smoother and you will reduce your editing time. You should be aware of overlapping new sized text and blocks with those that have not been scaled and, therefore, you should be prepared to do some manual editing of the "cleaned" drawing.

### Zero Elevation

Another common problem we have in day-to-day calculations in our surveying practice is dealing with 3D objects from architectural or engineering digital drawings and the preparation of digital files for them. It is straightforward to change elevations for points, texts and other objects positioned by one point. However, it is complicated to try to create projections for other objects like lines or 3D polylines. Therefore, the solution in this case would be to call "change

**Figure 2**

```
Sub TextSelection_Example()
    Dim SSetText As AcadSelectionSet
    Dim groupCode(0 to 1) As Variant
    Dim groupData(0 to 1) As Variant
    Set SSetText = ThisDrawing.SelectionSets.Add("SSetText")
        groupCode (0) = 0
        groupData (0) = "Text"
        groupCode (1) = 8
        groupData (1) = ThisDrawing.ActiveLayer.Name
    SSetText.SelectOnScreen groupCode, groupData
End Sub
```

In fact, the procedure of text scaling will work under the loop and may be like this:

```
Sub TextScale_Example(SSetText As AcadSelectionSet)
    Dim Entry As AcadObject
    Dim objText As AcadText
    Dim NewHeight as Long
    NewHeight = 1
    For Each Entry In SSetText
        Set objText = Entry
        objText.Height = NewHeight
        objText.Update
    Next Entry
End Sub,
```

where "NewHeight" is a function of the plan scale value. For example, for a plotting scale of 1 : 200 "NewHeight" can be 0.5, which is one quarter of the numerical scale factor 2 (200/100).

objects" or "creating projection objects", as in the following example.

When the object is a line, I have created a procedure that processes the coordinates of a starting point and an end point. Here is an example:

```
Sub LineCase_Example()
    Dim objLine As AcadLine
    Dim startPoint(0 To 2) As Double
    Dim endPoint(0 To 2) As Double
        startPoint (0) = objLine.startPoint(0)
        startPoint (1) = objLine.startPoint(1)
        startPoint (2) = 0
        endPoint (0) = objLine.endPoint(0)
        endPoint (1) = objLine.endPoint(1)
        endPoint (2) = 0
    objLine.startPoint = startPoint
    objLine.endPoint = endPoint
    objLine.Update
End Sub
```

The same operation will be successful for an arc or circle with the only difference being that we have to deal with the center point.

It is worthwhile noting that 2D polyline objects have a property called "Elevation". Changing the value of this property to zero will give us the desired effect.

The most difficult line types to deal with are 3D polylines because each vertex has a different elevation (or Z-coordinate).



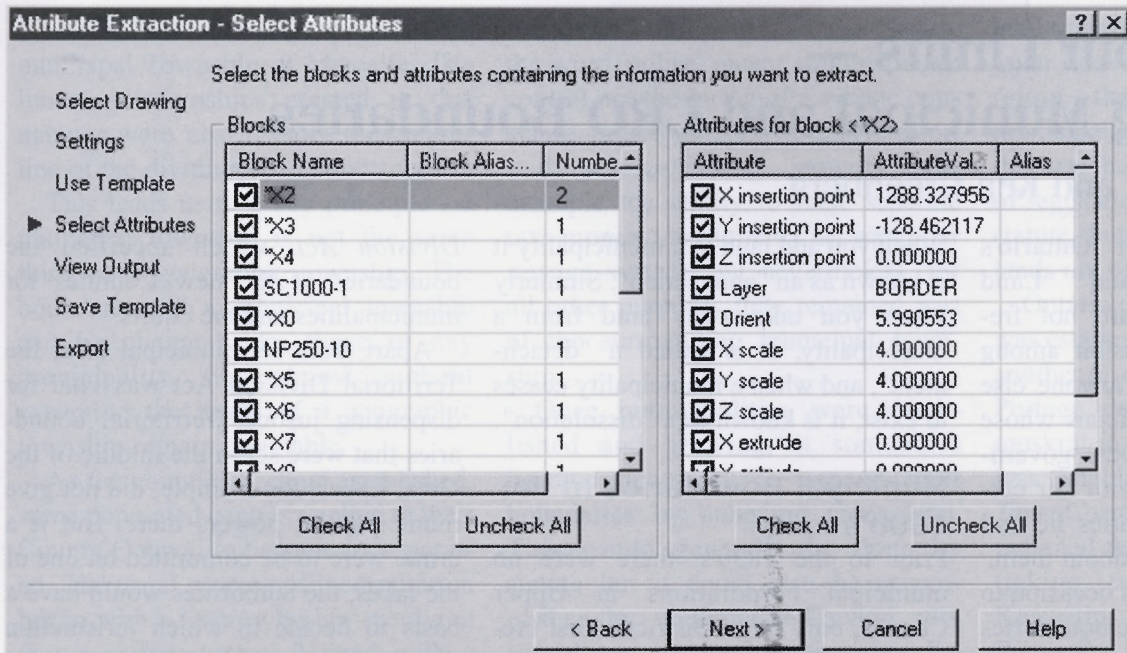


Figure 3

My experience has shown that it is possible to have 3D polylines in architectural, engineering and drawings prepared by photogrammetric methods. To create projections, we should write a loop procedure that will read coordinates of each vertex, change the Z value to zero and create a new 2D polyline with the correct vertex list. This would be a topic for discussion in a future paper.

### ASCII text format files

One of the most challenging problems we face is automatically extracting coordinates from AutoCAD drawings and creating ASCII files for use in field operations. Moreover, we are normally using some procedures created inside or outside AutoCAD. It is very simple to extract coordinates as attributes from blocks, inserted in our drawing using the existing built-in feature available in AutoCAD. The command can be found under the Tools Menu named "Attribute Extraction". For AutoCAD's command line, we may write the command "EATTEXT". This will lead us to a simple and friendly wizard that will help to create an ASCII file from block attributes (refer to **Figure 3**).

In the Blocks list, we can select the blocks that have attributes we want to extract and in the Attributes list, we can select the attributes we want to extract.

In the next step, we can choose between four available formats for the file:

- CSV - comma delimited ASCII file;
- TXT - tab delimited ASCII file;
- Microsoft Excel format file;
- Microsoft Access Database file.

All of these formats allow us the flexibility to manipulate the ASCII file we have created. In my opinion, it is more useful to deal with a database format that allows us to insert data stored as attributes (in AutoCAD blocks) directly in the database with specific columns and their aliases. This would be another topic for discussion in a future paper.

### Conclusion

In this article, I have defined some functions that can help simplify and automate routine work in the surveying office. My experience working with surveying companies around the world has shown me that a lot of people want to employ only the basic version of AutoCAD. It is ideal to be able to create small procedures dealing with problems that an office is working on, and having the option to customize the software to give us the fastest and most economical way to do so.



### Acknowledgements

I would like to thank G.G. Aldworth, OLS, M. Chapman, OLS, and B.J. Clarke, OLS, for proof reading the manuscript.

### References

1. Autodesk product discussion groups at <http://discussion.autodesk.com/>.
2. Boris Gutkin "GeoMenu: Automation of Geodetic Measurements", Point of Beginning Online Magazine, August, 2000.
3. Boris Gutkin "Technical condition for creating cadastral plans using AutoCAD for GIS". Izvestiya Vusov, Moscow, Number 3, 2001.
4. Jeffrey E. Clark "VBA for AutoCAD: writing AutoCAD macros". Upper Saddle River, NJ 07458.
5. VBDesign pages at <http://vbdesign.hypermart.net/cadpages/>.

**Boris Gutkin received his MSc in Geodesy in 1989 in St. Petersburg, Russia and his PhD in Geomatics in 2002 from Moscow State University of Geodesy and Cartography. He is currently working at A.J. Clarke and Associates Ltd. as a Geomatics Specialist and is articulated to Barry Clarke, OLS, OLIP. If you have any comments or questions, please do not hesitate to contact Boris at [bgutkin@ajclarke.com](mailto:bgutkin@ajclarke.com).**