Getting Started with Ivan

What is Ivan?

Ivan is a .NET class library written in C# that allows you to incorporate the use of AutoCAD® DWG® files into your .NET applications. It currently supports versions R15 (R2000), R18 (R2004), R21 (R2007), R24 (R2010) and R27 (R2013). Ivan consists of a single .NET assembly and is very lightweight and easy to use.

You can use Ivan to load DWG® files then access all the contents of those files including objects, entities, sections, xdata, xdictionaries, xrecords, etc. Essentially, the entire contents of the drawing file are at your disposal. As the drawing is loading, Ivan checks all Crcs, Checksums and Check Data within the file to ensure its integrity and issues an exception if the file is corrupt.

How to use Ivan

In order to use Ivan you set a reference to the Ivan assembly in your application project. Then you instantiate a Ivan.Drawing object and call Read, at which point the Drawing object will provide total access to the entire DWG® file contents. Listing 1 gives an example of this.

```
using Ivan = ComponentIngenuity.Ivan;

Ivan.Drawing objDrawing;

objDrawing = new Ivan.Drawing();
objDrawing.Read(@"c:\location\mydrawingfile.dwg");
```

Listing 1. *Creating a Ivan.Drawing object.*

Once the Read method has been called the Drawing object is initialized and ready for use. The contents of the Ivan object model mirror those of the internals of the DWG® file. The top level objects are accessed via lists and dictionaries. The DWG® symbol tables are exposed as typed lists and DWG® dictionaries are exposed as dictionaries, with a few also being exposed as lists, as a convenience.

Listing 2 gives an example of how you might obtain a list of all the Layers in the drawing and place them in a ListView.

```
ListViewItem objItem;
int i;

//
// using a for loop
//
for (i = 0; i < objDrawing.Layers.Count; i++)
{
   objItem = new ListViewItem(objDrawing.Layers[i].Name);
   listViewLayers.Items.Add(objItem);
}

//
// using a foreach statement
//
foreach (Ivan.Layer objLayer in objDrawing.Layers)
{
   objItem = new ListViewItem(objLayer.Name);
   listViewLayers.Items.Add(objItem);
}
```

Listing 2. *Listing the Layers in a drawing.*

Accessing the entities

All of the top level objects like Blocks, Layers, Layouts, Linetypes, etc. are accessible through their associated drawing list property. Entities are different. For the most part all entities are accessible via their parent Block object. For example, the most fundamental element in a DWG® file is the Model Space Layout object, which houses all the entities that define the drawing's geometric properties. To access these entities you would use the Model Layout's associated Block object.

Unlike the top level objects, which are readily available, in order to access the entities you must query for them using the Ivan.EntityDatabase Select method. Each Block object has a Entities property typed as a Ivan.EntityDatabase object. The EntityDatabase object exposes a Select method which is overloaded to allow you to access the entities in a number of ways.

The Select method is designed in such a way as to return the result set as a page of entities. After the initial Select method is called you can page through the result set, incrementally, to access the entire set of entities. You can specify the size of the result set page size or indicate to return all. In addition you may also limit your results to a specified list of Layers and/or entity types, like Lines, Circles, Text, etc.

You can use the *IvanReference.chm* file to explore all the possibilities, but Listing 3 gives an example of how you would access all the Text items on the “HVAC” Layer. A page size of zero is specified to return all.

```
Ivan.DrawingEntityList listEntities;
Ivan.Layout objLayout;
List<string> listLayers;
List<Ivan.ObjectType> listTypes;

objLayout = objDrawing.Layouts["Model"];

listLayers = new List<string>();
listLayers.Add("HVAC");

listTypes = new List<Ivan.ObjectType>();
listTypes.Add(Ivan.ObjectType.Text);

listEntities = objLayout.Block.Entities.Select(Ivan.FilterMethod.Inclusive,
listLayers, Ivan.FilterMethod.Inclusive, listTypes, 0);
```

Listing 3. *Accessing all Text items on the HVAC Layer.*

The IvanExplorer sample application

The best way to get acquainted with the Ivan object model and how to use it is to examine the *IvanExplorer* sample application along with the *IvanReference.chm* documentation file. The trial download will provide you with both of these. The trial comes with, both, a ready to run binary of the sample application, as well as the complete Visual Studio project and source code of the sample.

By examining the sample source code and the Ivan API reference document you can get up to speed, rather quickly, on what the object model consists of and how to use its elements. The Ivan API reference will outline all of the objects and properties and the sample application will display them in a dialog box with all of the property values as they are taken from the drawing.

The sample is pretty exhaustive and provides a usage example of practically every object and property in the Ivan API.

If you are new to the AutoCAD® DWG® file format a good way to learn about the inner workings of the format is to download a copy of the DXF® specification. It will provide you with knowledge of all the object and entity types as well as all of the terminology that is normally used in this context. Once you have absorbed that you will begin to recognize the relationship between the Ivan object model and the DWG® file format.