Title: A .NET ArcIMS Wrapper for the Rest of Us

Author: Justin White

Abstract: One of the challenges of effectively using GIS in any organization is the fact that GIS is a highly specialized technology with its own set of technologies and terminologies. As a result, in order to GIS enable an application, one has to be proficient in ESRI technologies which are a key barrier to widespread adoption of GIS within an organization. To break down the barrier, we have to make GIS tools more readily available to our application developers. To that end, we have designed a .NET component which hides most of the implementation details from the developer. .NET developers will simply drag a map component on to their Web page, add the layers they need and wire up code to user events such as a map selection. Because this component relies on a standard ArcIMS Service, a developer new to GIS can create a fully functional GIS application in minutes.

Paper Body:

At the City of Eugene in Eugene, Oregon, the Geographic Information System is maintained by a division under the Public Works department. This division is currently responsible for creating and maintaining the data that makes up the city's GIS, but does not provide programming resources. Programming resources for each city department work as separate groups under the Information Services Division (ISD). Of these seven groups of analysts, I am the only one, working for Public Works who was assigned to GIS programming projects for no more than 70% of my time.

Since the ISD work groups are dedicated to each department, cross departmental collaboration on IT projects are rare. The city was having a tough time leveraging the investment they had made in ArcIMS 9 infrastructure. Although customers, both internally and externally, are asking for GIS enabled web applications, Public Works is the only department with an analyst who has the training necessary to create GIS enabled web applications.

Like many municipalities, the city was in a budget crunch. Money for GIS proficient staff or GIS training for existing staff is hard to come by. But in order to meet customer needs, something had to be done to quickly give analysts working for departments other than Public Works the ability to create GIS enabled web applications. To meet this goal, the city explored a range of options that would allow development of web applications using ASP.NET, the city's web platform of choice.

Training was an obvious choice, but as mentioned above, money for training was just not available. To further complicate training options, GIS, like any specialized technology, has its own set of terminology and technologies that cannot be learned overnight.

It quickly became apparent that an Application Programming Interface (API) was needed which would hide the complexities of a GIS system. At the same time, the API should allow a developer to create sophisticated applications capable of performing common GIS tasks such as zooming, panning, selecting and buffering. It was decided that this API should be integrated with Visual Studio.NET (VS.NET), preferably as a visual component allowing the user to drag and drop tools onto a web form like other components. In addition, this API would need to integrate seamlessly with the Plumtree portal the city was in the process of implementing.

Once the decision was made that a GIS API or component was most likely the solution, the search for one began. The first two options investigated were the ActiveX Connector and the new .NET Link that shipped with ArcIMS 9. Unfortunately, the ActiveX connector lacked any type of visual interface, and did not integrate with VS.NET. At first glance, the .NET Link appeared crippled and almost useless. The city investigated purchasing a solution, but unfortunately vendors were few and far between, and we would need the capability to customize the source code so that any GIS applications could integrate with the portal.

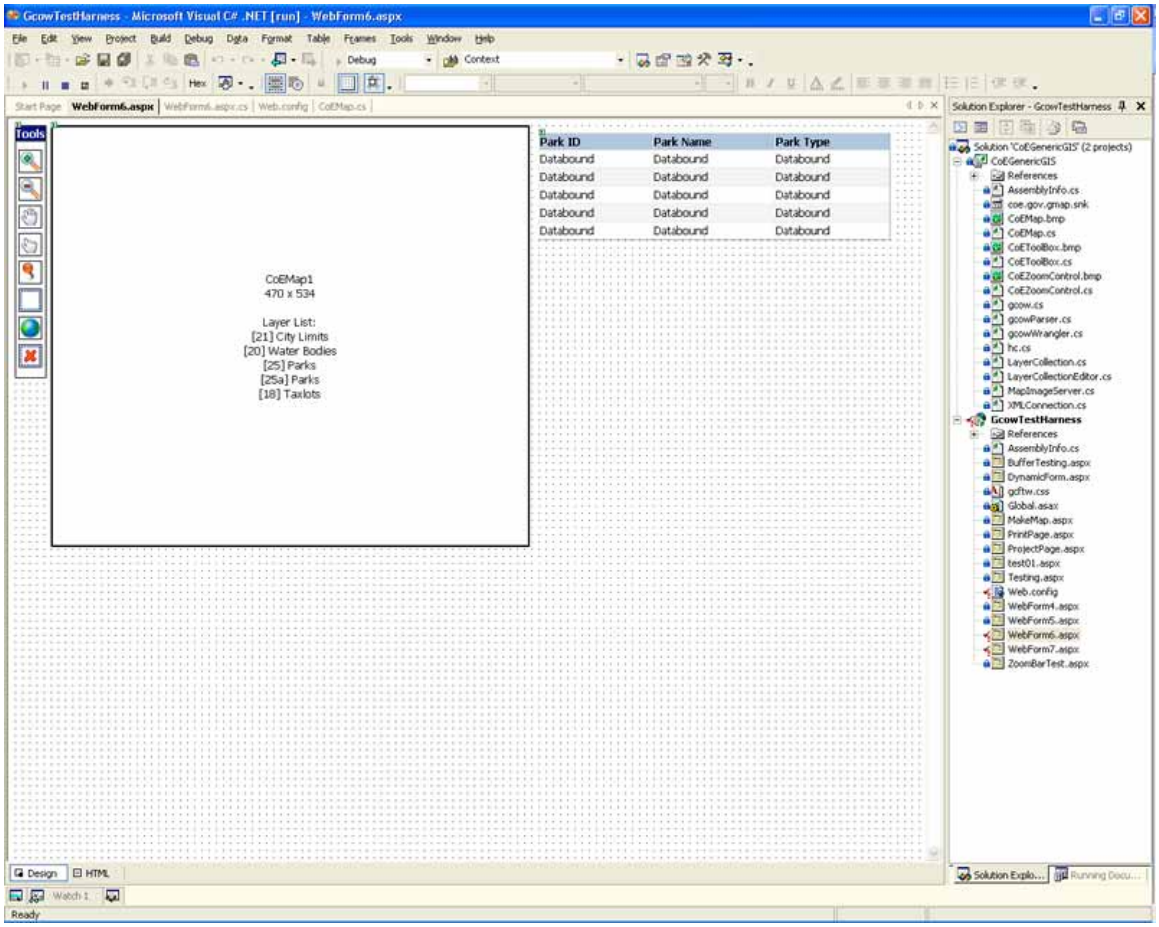
We concluded that we needed to build our own component to satisfy our list of requirements. In October of 2004, I began developing gMap (Generic Map), a .NET ArcIMS Wrapper for the Rest of Us. Eight months of part-time development later, gMap was successfully used in the city's first publicly available GIS enabled web application written by an analyst working for the Police department.

Based entirely on ESRI's .NET Link, gMap is made up of one dll with ~6500 lines of c# code, and ~1200 lines of JavaScript code. It is fully integrated with Visual Studio.NET, is developer friendly and supports most major browsers. gMap is also portal "aware" - it can recognize when it is running under the context of the Plumtree portal and seamlessly gateway requests to and from the portal servers.

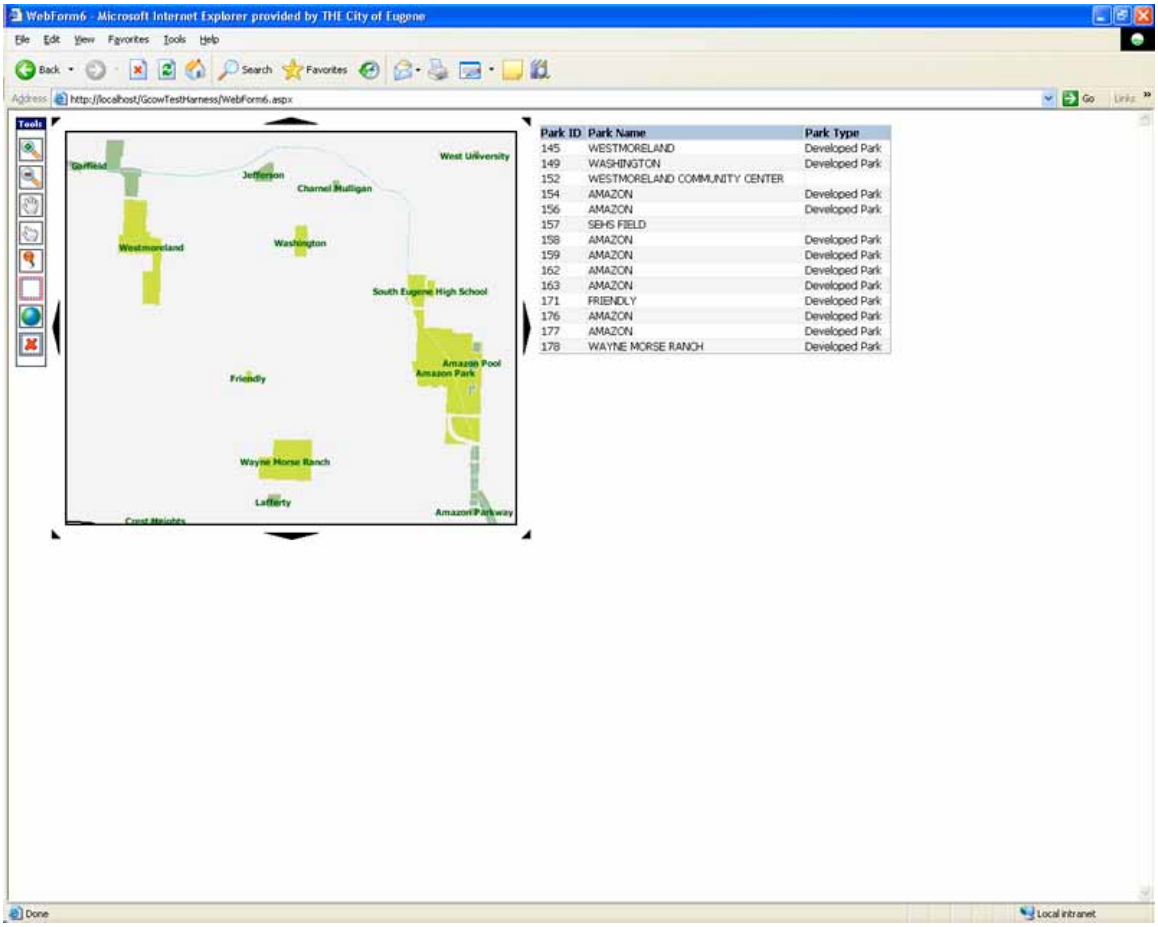
gMap has three visual objects inside of VS.NET, the map, toolbox and zoom bar. The map contains logic to display information to the end user as well as the ability to handle user interactions. The toolbox contains common tools found in a typical GIS application including zoom in, zoom out, pan, center, identify, box select, and zoom to full extent and clear selections. The zoom bar allows the user to zoom the map in and out a fixed number of steps.

gMap's API allows a developer to perform selections, queries and buffers against the GIS data as well as control the appearance of the final image. In its default mode, gMap works against a single generic or custom service containing all layers deemed acceptable for public consumption. Adding layers to a map is handled through a "pick list" generated on the fly from the ArcIMS service gMap is pointing to. This feature allows developers not familiar with city data to quickly choose the layers they need in their application.

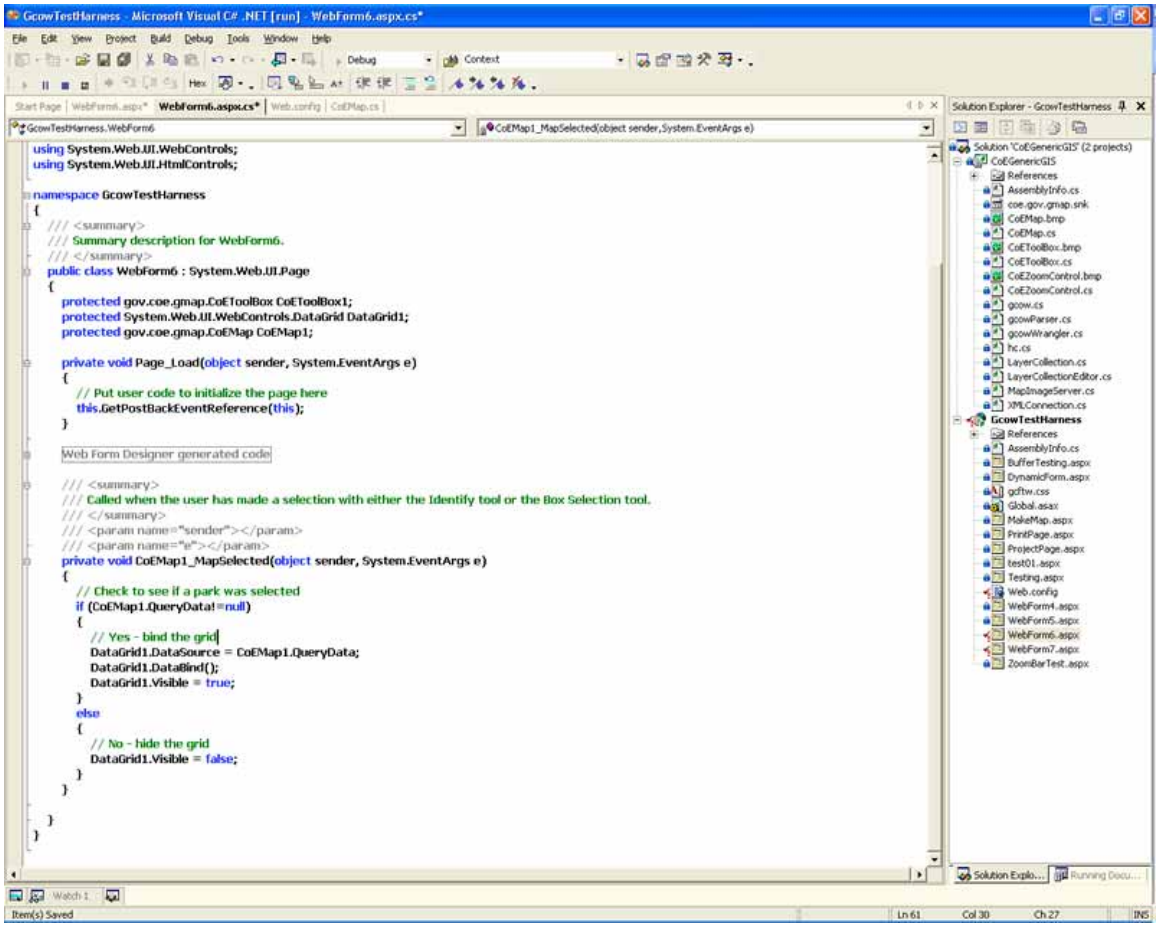
To see gMap in action and find information on its' release status please visit the City of Eugene at http://www.eugene-or.gov



gMap's map and toolbox objects inside of Visual Studio.NET



The above application running in Internet Explorer. The user has selected a section of parks using the box selection tool. The results are displayed in the Datagrid on the right.



The code behind the simple application demonstrated above.

Acknowledgements: I would like to thank ESRI for the .NET Link as without it this project would not be possible. I would also like to thank Peter Shum, Barry Bogart and Jan Wostmann for their help throughout the development of this project.

Appendixes: None

End Notes: None

Author Information:
Justin White
Sr. Systems Analyst
City of Eugene
100 W. 10TH Ave, Eugene OR 97401
P: (541)682-5857
F: (541)682-6899
justin.b.white@ci.eugene.or.us