

Extending ArcIMS with PHP and MySQL

Byron Cochrane
Senior GIS Analyst
Ada County Assessor's Office GIS

Paper Abstract

The Ada County Assessor Office has brought the power of GIS to a broader audience by incorporating the open source tools PHP and MySQL with ArcIMS. The two applications to be examined are a parcel map viewer attached to an AS400 Property Information System and Adapar, an Internet parcel mapping system linked to tabular data stored in MySQL. The first example will show how the spatial components of MySQL can be harnessed to improve performance of a simple map viewer. The second example will illustrate how Ada County harnessed the power of PHP and MySQL to provide quick access to tabular characteristics of parcels viewed on an ArcIMS internet implementation. Used together, PHP and MySQL can provide a powerful, rugged, well tested, and inexpensive solution for improving performance and extending the capabilities of an ArcIMS implementation.

Introduction

The Ada County Assessor's Office GIS has been very successful implementing ArcIMS-based internet and intranet sites built in part on two open source project: PHP and MySQL. Once installed, they have provided a powerful, rugged, well tested, and inexpensive solution for improving performance and extending the capabilities of ArcIMS. These mainstream tools have enabled Ada County to have a fast, efficient web site that integrates ESRI spatial data and AS400 tabular data. This paper will describe two ways that we used these open source tools in conjunction with ArcIMS to both extend the capabilities of ArcIMS and to integrate our GIS more fully with the functions of the Ada County Assessor's Office.

Ada County GIS was started in 1986 when a private company developed this GIS on spec. It was later acquired by Ada County. By leveraging a tight control network and years of cooperative effort with other local entities, by the year 2000 our GIS had reached a high level of accuracy and currency. However, being a database of coverages stored in ArcInfo Librarian, it was easily accessible only to local GIS professionals.

Ada County Assessor Robert H. McQuade wanted to make information available to the majority of potential users who only wished to view and query this data. The recently developed ArcIMS became our tool of choice to accomplish this. It provided centralized administration and gave the end user an ease of use to that was not available in other tools.

In our first ArcIMS effort, we created two largely identical websites: one Internet with some limitations to information, and one Intranet that provided sensitive information such as owner name and address. Both sites were created using a modified version of the ESRI supplied html interface.

Users of both the Internet and Intranet wanted more sophisticated access to our AS400 database's assessment information. Simply attaching more data to the shapefiles did not provide the performance we desired, so we initiated a search for a better solution. We decided upon was a combination that has been popular with internet application developers creating database driven websites for years: the open source tools of PHP and MySQL.

The recursively named PHP (PHP: Hypertext Preprocessor), a project of the Apache Software Foundation, is a widely used general purpose scripting language that is especially suited for Web development and can be embedded into HTML. One of its strengths is the ability to connect to a large number of databases. It is often used in tandem with MySQL.

MySQL is the most popular open source relational database management system available. It is heavily used in many web applications. It is developed, distributed, and supported by MySQL AB of Germany. Version 4.0.1 of MySQL introduced spatial extensions allowing for the generation, storage, and analysis of geographic features.

Two Applications Extended

We implemented our first ArcIMS application in the spring of 2000 using ArcIMS 3.0. It was a parcel viewer intended to be used by the public to check if their assessed values were comparable to similar properties. One could search for property through a geocoded centerline file or by parcel number. Other layers, such as streets, water, city limits, etc., were incorporated for reference, but selections and identifies were limited to the parcels layer.

This first ArcIMS effort was built using modified versions of the html and javascript code shipped with ArcIMS. Because of performance and other considerations, the data attached to the parcel shapefile were limited. Important characteristics had to be left out due to complexity of the shapefile's dbf. Our own appraisal staff and outside users requested that I find a way to get beyond this limitation.

In addition, it was desired that we provide some basic access to our GIS data through our Property Tax System (PTS). Developed in-house, PTS provides thick client access to our AS400 database. Our appraisers and others inside Ada County use PTS to view and edit tabular property information.

My background is neither web design nor relational databases. Although I had by this time learned something of Javascript, HTML and other languages associated with maintaining an ArcIMS site. I am an old AML programmer by history. Due to limited personnel resources I needed a simple, reliable solution that I could maintain and support myself. This search led me to PHP and MySQL. Following the online guides, I was able to install and run precompiled versions of these open source software with relative ease.

Two tasks that proved to be somewhat more challenging were programming and data maintenance. Programming in PHP is not too difficult to learn, as it is a very flexible language. It can be embedded in html and javascript and both of these can be embedded in it. And since the tasks that I required of it were limited, I was able to find similar scripts that I was able to hack into a form that I could use.

As of version 4.0.1, MySQL implemented spatial extensions following the specification of the Open GIS Consortium (OGC). I will show how I used this to my advantage in my second example.

Installing and Managing PHP and MySQL

MySQL is downloadable from <http://www.mysql.org>. It is available for almost any platform. We have installed versions on Windows, AIX-Unix, and linux. Binaries are available for most platforms. On windows, a standard installation wizard loads the program. A single command loads the program as a service. Currently, we use versions MySQL 4.0.20 and 4.0.17. Various html interfaces are available for managing MySQL databases. We have used a few different interfaces including PhpMyAdmin (www.phpmyadmin.net). We have also created our own VB programs and batchfiles of basic MySQL commands to manage our database. I will describe these tools in more detail later.

PHP is freely available from <http://www.php.net>. There are versions for most platforms in both source code and binaries. We have installed PHP on Windows, AIX-Unix, and linux. A downloadable installation script configures IIS, PWS and Xitami for PHP. Straightforward manual configuration instructions exist for other platforms. We use version PHP4.3.7 with Apache 2.0.48. Some additional customization of two files and a modification to apache.conf is all needed for PHP to function on windows. PHP documentation describes the installation of PHP4 with Apache2 as experimental and has called it such for years. However, we have neither experienced nor heard of any experiences that leads us to mistrust this combination.

Example 1 – A Database extension to ArcIMS Application

An early identified shortcoming of our first internet site was that it was difficult to include related appraisal characteristics for a selected parcel. These characteristics tables reside on an IBM AS400 and are administered through custom applications. Our initial method of showing property characteristics was to download tables via ODBC connections (using Showcase Strategy) and join the resulting DBF tables to a shapefile. While this method allowed ArcIMS to display these data, it was found to be a cumbersome and slow approach in our county with its 135,000 parcels with millions of associated records. The logical solution was to minimize the data in the dbf files of the shapefiles and to provide links to a separate database when a parcel number was selected.

One advantage that I had in my existing deployment was that I had limited queries to a single layer: parcels. This allowed me to alter the javascript in aimsIdentify.js to add hotlinks to specific named fields and values of the parcel shapefile. When the identify tool was used a popup window would appear showing some basic characteristics attached to the shapefile. I then hotlinked some of these characteristics fields to MySQL tables via PHP. These hotlinks were made apparent to the user by the default color of blue. Clicking a hotlinked field triggered a PHP script that passed two values—the parcel number and table name—to the MySQL database using the GET_METHOD (see below). The returned records were then parsed to populate a new popup window of detailed parcel characteristics. The major advantage of this approach was speed; in addition, the characteristics could be updated without restarting any services.



One of the main hurdles we encountered in implementing this solution was data maintenance. A shortcoming of MySQL is that it does not ship with a very robust graphical user interface for administration. There are, however, many tools commercially and freely available for this task. We chose PhpMyAdmin for some of these tasks. While it does not provide tools for automation of data loading, it does provide good tools for jobs such as indexing, editing schemas, and managing users and permissions.

Before data can be loaded into MySQL a schema needs to be created. This can be done manually through MySQL command line or a bit more easily through PhpMyAdmin . There are also third party tools that can be purchased or download such as MyDbConverter <http://www.mydbconverter-access-mysql-converters.com-download.net/> that copies data along with schemas from other databases into MySQL. One converter that we have used with limited success is DBF2MySQL, which is a non-supported project on <http://www.sourceforge.net/>. More recently we have created our own VB tool, which recreates and populates data from AS400 into tables in MySQL. Once the tables are created, updating them from text files is a relatively easy task through simple batchfiles.

We were already on a weekly update cycle for the parcel shapefile. This shapefile is created from an ArcInfo library layer utilizing regiondissolve to create unique regions for each parcel number. These regions are then converted to a polygon shapefile, and the associated data is attached.

With our new approach, the MySQL characteristics tables also need to be updated weekly. Using Showcase Strategy, a tool we already possessed, we extract the needed data via ODBC from the AS400 into separate text files and then load these data into MySQL with the above mentioned batchfiles.

One illustration of the sturdy and forgiving nature of MySQL is in how I move this updated data to the website. Our Internet server has no connection to our internal network, so I create the new MySQL database on a separate but networked computer. Under the MySQL data directory I locate the folder with the name of the database I just updated. I copy and paste this folder onto a Jaz drive. Then I transfer this drive to the Internet computer and drag and drop this folder into the MySQL data directory on that machine. The Internet services instantly see the new data without refreshing any services, and with no apparent performance penalty. That is what I call idiot-proof!

Example 2 –An ArcIMS Extension to a Database Application

The other task of integrating ArcIMS into our Property Tax System required an almost opposite approach. The PTS already incorporated photographs and documents. Now it was desired that it incorporate GIS. The view provided by Internet parcel map would suffice, but it needed to focus on the parcel being detailed. Launching the full website to look at one parcel would be excessive. Rather it was desired that the map launch in a window similar to the document window, a couple of hundred pixels square.

In the ArcScripts section of the ESRI Support Center we discovered MicroMap, a very small, (about 12K) and very simple frameless HTML viewer. It uses the jpost applet for communication. This tool provided us the lightweight viewer we needed for the intranet. Through the rapid application development tool used to create the PTS, we were able to add a right-click option to launch an Internet Explorer window with a URL. Through a javascript hack, we were able to get this window to open at the desired size and, to clean up the appearance, without any Internet toolbars.

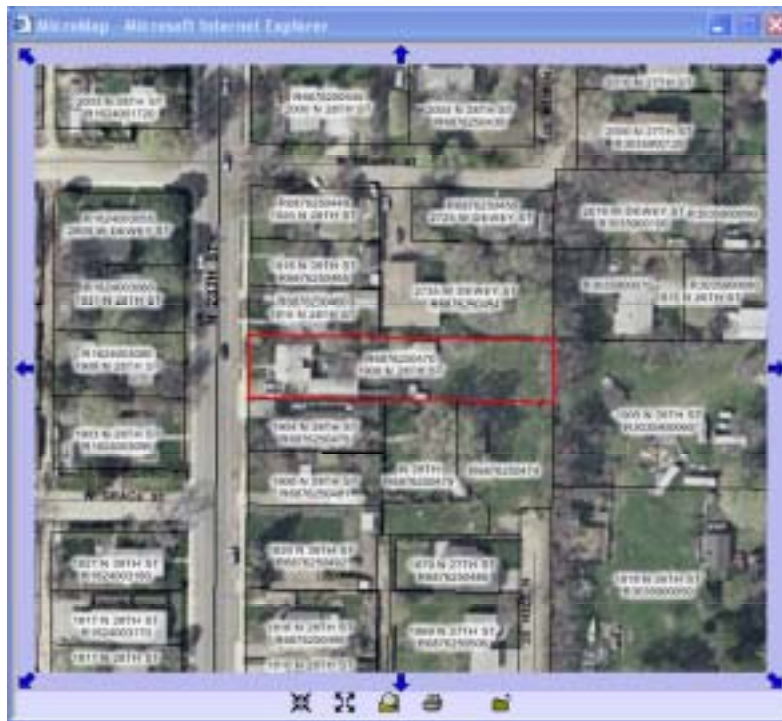
The URL connected to the parcel ArcIMS mapservice via MicroMap. The parcel number included in the URL is retrieved with the GET_METHOD in PHP. This number is then passed into an AXL function that retrieves the extent of the parcel, selects the parcel for highlighting, and draws the appropriate map on the screen. This “worked”—but was extremely slow and was therefore not implemented.

The introduction of spatial capabilities in MySQL 4.0.1 allowed us to speed up parcel extent extraction. The tool toWKT interface extension for ArcView 3.2, an application that is part of the GeoServer Project at <http://geoserver.sourceforge.net/>, provides a way to load shapefiles into MySQL. This is still the tool we are using today, but because it is limited to the loading of shapefiles in ArcView 3.2, we are in search of more flexible tools.

The parcel shapefile now residing in MySQL included five spatial attributes, Geometry, Xmin, Xmax, Ymin, and Ymax. In addition, we included the parcel number as a foreign key. Through PHP a parcel could then be selected from the MySQL database. Once a parcel was selected we could extract its X and Y extents, pass these coordinates into an AXL GET_IMAGE request, and then draw the map at the appropriate scale. This improved the performance greatly, but not to the level we desired.

We still needed to select the parcel to highlight it—a slow process on a shapefile of 135,000 elements. This time, the Geometry field in MySQL provided the solution. Through much trial and error, I discovered I could extract the contents of the Geometry field (stored in a LONGBLOB) and parse out a usable set of coordinates. These coordinates were passed to an AXL GET_IMAGE function that would then draw an acetate layer on the MicroMap view every time it is refreshed. This has the same effect as highlighting the parcel in ArcIMS, but shaves many seconds off the MicroMap display time. It is now a usable viewer that after initial loading is quite

fast. (The slowness at initial load is due to the time it takes to load the MicroMap applet into the browser. Subsequent views are very fast.)



Conclusion

How long we will continue the PHP and MySQL approach is in question. Recently we migrated our web sites to Geocortex IMF from Latitude Geographics of Victoria, British Columbia. We gained a more robust, stable, and responsive HTML viewer. The package utilizes java server pages to create the viewer and connects to ArcIMS map services using the java connector. I have adapted their JSP to call the same PHP scripts that I used in the previous version of our site. To simplify code maintenance, we are now replacing the PHP code with JSP to extract data from MySQL. Where the option exists, we are also exploring the possibility of extracting the data directly from the AS400 to the website; this would eliminate the need for MySQL.

Like many other GIS shops, we are in the process of converting our database to SDE Geodatabase. The RDBMS we have chosen is Microsoft SQLServer. This move will have many effects on our web site designs. MS SQLServer will likely make MySQL redundant. Also, the performance gains we experienced by leveraging the spatial components of MySQL may be unneeded when we have access to ArcSDE.

Still, the advantages of using PHP and MySQL are great. Once installed it provides a powerful, rugged, well tested, and inexpensive solution for improving performance and extending the capabilities of ArcIMS. These open source tools are well suited for the web and are heavily used in that environment. Each has a large community of users for support. The stability of these projects' development is furthered by the participation of major corporations in their development, such as SAP (in the case of MySQL) and IBM (in the case of the Apache Foundation which oversees the development of PHP). These are mainstream tools. If your goal is to integrate IMS into the broader world of the Internet services PHP and MySQL can be useful tools to have in your toolbox.

Acknowledgments

Special thanks go out to Robert H. McQuade, Ada County Assessor, for having the vision to see the value in this effort. Also, thanks go out to Sheldon Bluestein, Land Records Supervisor, for making available all the necessary resources, and for the great help he gave editing this paper. And finally, thank you Randall Svancara, GIS Technician, for nudging me in the right directions when I was floundering in new programming environments.

Appendices

Attached files:

MyChar.php

Used in parcel viewer to retrieve data from MySQL and display results.

MicroMap.php

Calls MicroMap2.php and resizes window with javascript.

Called by external program. Passes a parcel number.

MicroMap2.php

A php program rapped around the ESRI ArcScript Micromap.js.

The php portion extracts geographic coordinates from MySQL data.

Author Information

Byron Cochrane
Senior GIS Analyst
Ada County Assessor's Office GIS
200 W Front St
Boise ID 83702
ascochbe@adaweb.net
208-287-7274