

## ArcIMS Implementation at the City of Berkeley June 15, 2003

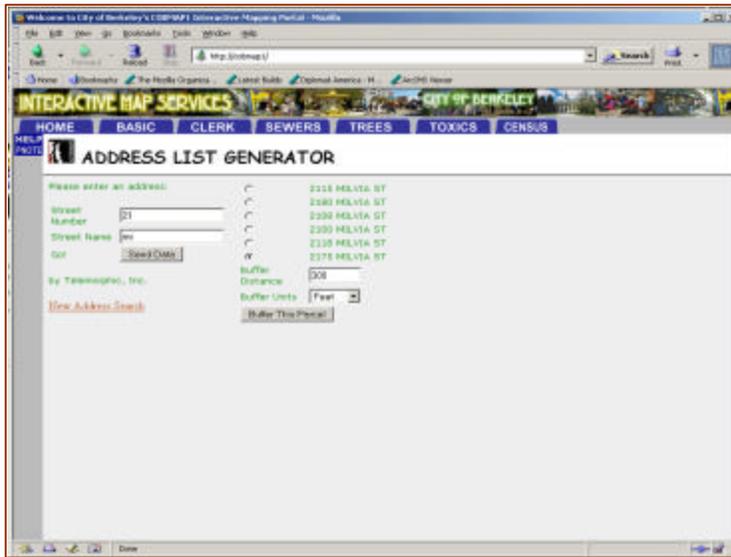
Brian Quinn, Application Programmer/Analyst - Enterprise Implementer

Building on earlier adoption of a related custom application, the City of Berkeley has undertaken an enterprise approach to distributing *feature* and *raster* data plus *spatial* services using ESRI's ArcIMS software. An essential aspect of our implementation is a geodatabase foundation where common legacy feature layers are provided defined projections and loaded in feature dataset pairs: California state plane coordinates in NAD83 feet for continuity with the past, and UTM NAD83 meters for US National Grid (USNG) integration and USNG feature labeling. Raster imagery from scores of orthophoto tiles, hundreds of georeferenced scanned documents, and various numerical models have been mosaicked and loaded as raster features in ArcSDE. Raster performance is consistently faster than local disk storage whenever the source imagery is large, and some of our mosaics are over 20 GB.

By IT Department policy we use ESRI software, Windows 2000 server OS, and MS SQL Server 2000. Our enterprise architecture has two dual-Xeon Dell servers, one mostly for ArcSDE, the other mostly for ArcIMS internal use. The Apache 2 web server and Tomcat 4 Java servlet engine were chosen for performance, reliability, and consistency with options for public-facing ArcIMS on Linux. No accommodation of MS Active Server Pages (ASP) has been made on our ArcIMS server, as that platform is supported on other city web servers running MS IIS and ASP. We only use browser plug-ins in a few cases where users need more dynamic analysis than is convenient to offer as a map service and also that user's department chooses not to use ArcView 3.3 or ArcGIS/ArcView 8.3. In these cases, we use the Telemorphic Maplicity (Java) plug-in. Map services are tested on Mozilla 1.3/1.4 and Internet Explorer 5/5.5/6.

Our GIS analysts and manager are able to create ArcMap documents drawing solely from ArcSDE imagery and features rather than any local files. Such map documents are easily published as ArcMapServer services, and can also be shared as attachments and opened without data path issues for recipients in view of our ArcSDE server. Any appearance constructed in ArcMap's data view is likely to be available through the corresponding ArcMapServer map service and viewed directly in ArcMap or through an HTML map service. A census data browser has been built this way and is quite popular with internal users. ArcMapServer services are a convenient test bed for quickly presenting new map service concepts to our internal users.

When a map service is popular enough to make faster performance important, we compose an ArcXML (AXL) file to define the service. While it is relatively more awkward to compose a layer's appearance or include raster in the AXL than to create a map document, the result is typically a faster service and better image quality and consistency of reception on large and small screens than we find with our ArcMapServer services. Several work processes have been automated with ArcIMS Designer-created services polished with minor DHTML adaptations to customize look and feel and provide simultaneous display of both legend and layer list. City Clerk's office staff can identify which council district a caller's address is within; engineers browse 1:600 sewer drawings seamlessly mosaicked citywide as 5-cm imagery; multiple years of 15-cm orthophotography may be browsed tile-free with street name annotation derived from centerline features; and land-use maps from the 1970's at 1:1200 scale have been brought online as a seamless 8-cm raster layer valuable for its feature detail and historical record. Other HTML map services range from toxics management sites and inventory to the Forestry Division's city tree management inventory.



Our most customized use of ArcIMS is a parcel notification application developed for us by Telemorphic and implemented in Java Server Pages (JSP) using the ArcIMS Java connector. An address fragment is entered and one or more likely actual address candidates are returned. Choosing an actual address, one can buffer from 0 to about 1500 meters (or comparable feet or miles,) from the corresponding parcel boundary, then compose lists of property (parcel) owners, occupants (address points, many to a parcel for apartment units),

or interested community organizations (defined by the groups in overlapping polygons) within the buffer around the chosen parcel. Lists are returned as comma-spaced values. We maintain this application internally using self-taught JSP knowledge grown on a structured C background.

Planning Department users put the Notifier to daily use, and our rare service interruptions have resulted from data issues such as updates to parcel polygons and address points. With ArcIMS we have continued to provide work process automation to our internal users and been able to decommission a MapObjects IMS server (very usefully converting over our Unisys GIF license in the process!) By choosing a JSP implementation, we leveraged the servlet engine already installed for ArcIMS, we avoided reliance on MS IIS or Sun ONE, and we found synergy in implementation with Telemorphic's Java expertise that has produced its Maplicity viewer.

Our Planning Department users are pleased with the application, and have asked for two variations already: one to display all the zoning and hazard areas that impact a given parcel, and a tool for listing all liquor licensees within a given radius to evaluate new liquor licenses. We anticipate that with relatively small additional effort, the parcel notification technology can be adapted to these and other work process automation requests.

