

# Microsoft TerraServer: A Spatial Data Warehouse

Tom Barclay

Jim Gray

Don Slutz

June 1999

Revised February 2000

Technical Report

MS-TR-99-29

Microsoft Research

Advanced Technology Division

Microsoft Corporation

One Microsoft Way

Redmond, WA 98052

# Microsoft TerraServer: A Spatial Data Warehouse

Tom Barclay  
Microsoft Research  
301 Howard St., Suite 830  
San Francisco, CA 94105  
415 778 8223

tbarclay@microsoft.com

Jim Gray  
Microsoft Research  
301 Howard St., Suite 830  
San Francisco, CA 94105  
415 778 8222

gray@microsoft.com

Don Slutz  
Microsoft Research  
301 Howard St., Suite 830  
San Francisco, CA 94105  
415 778 8226

dslutz@microsoft.com

## ABSTRACT

Microsoft® TerraServer stores aerial, satellite, and topographic images of the earth in a SQL database available via the Internet. It is the world's largest online atlas, combining eight terabytes of image data from the United States Geological Survey (USGS) and SPIN-2. Internet browsers provide intuitive spatial and text interfaces to the data. Users need no special hardware, software, or knowledge to locate and browse imagery. This paper describes how terabytes of "Internet unfriendly" geo-spatial images were scrubbed and edited into hundreds of millions of "Internet friendly" image tiles and loaded into a SQL data warehouse. All meta-data and imagery are stored in the SQL database. TerraServer demonstrates that general-purpose relational database technology can manage large scale image repositories, and shows that web browsers can be a good geo-spatial image presentation system.

## Keywords

Geo-spatial, VLDB, image databases, internet.

## 1 Overview

The TerraServer is the world's largest public repository of high-resolution aerial, satellite, and topographic data. It is designed to be accessed by thousands of simultaneous users using Internet protocols via standard web browsers. TerraServer is an image "tile" server that delivers a set of raster images based on a users search criteria. Once an image of interest is located, users can pan, zoom in, zoom out, or display meta-data about the image they are viewing.

The TerraServer is a multi-media data warehouse. It differs from a traditional data warehouse in several ways: (1) it is accessed by millions of users, (2) the users extract relatively few records (thousands) in a particular session

and, (3) the records are relatively large (10 kilobytes). By contrast, classic data warehouses are (1) accessed by a few hundred users via proprietary interfaces, (2) queries examine millions of records, to discover trends or anomalies, (3) the records themselves are generally less than a kilobyte. In addition, classic data warehouse queries may run for days before delivering results. Initial results typically cause users to modify and re-run queries to further refine results.

One thing the TerraServer has in common with classic data warehouses is that both manage huge databases: several terabytes of data. TerraServer's topographic maps cover all of the United States at 2 meter resolution (10 million square kilometers), the aerial photos cover 40% of the United States today (3 million square kilometers) at one-meter resolution, and 1% of the urban areas outside the United States (1 million square kilometers) at 2 meter resolution.

This report describes the design of the TerraServer and its operation over the last 18 months. It also summarizes what we have learned from building and operating the TerraServer.

Our research group explores scaleable servers. We wanted first-hand experience building and operating a large Internet server with a large database and heavy web traffic. To generate the traffic we needed to build an application that would be interesting to millions of web users.

Based on our exposure to the EOS/DIS project [2], we settled on building a web site that serves aerial, satellite, and topographic imagery. We picked this application for four reasons:

1. The web is inherently a graphical environment, and these images of neighborhoods are recognizable and interesting throughout the world.
2. We believed this application would generate the billions of web hits needed to test our scalability ideas.
3. The data was available. The USGS was cooperative, and since the cold war had ended, other agencies were more able to share satellite image data. The thaw relaxed regulations that had previously limited the access to high-resolution imagery on a global basis.
4. The solution as we defined it – a wide-area, client/server imagery database application stored in a commercially

---

© ACM 2000. This article appeared in the Proceedings of the ACM SIGMOD, May 2000 in Austin, TX. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

available SQL database system – had not been attempted before. Indeed, many people felt it was impossible without using an object-oriented or object-relational system.

This paper describes the application design, database design, hardware architecture, and operational experience of the TerraServer. The TerraServer has been operating for eighteen months now. We have deployed the third redesign of the database, user interface, and process of adding new images to the database.

## 2 Application Design

TerraServer is accessed via the Internet through any graphical web browser. Users can zoom and pan across a mosaic of tiles within a TerraServer scene. The user interface is designed to function adequately over low-speed (28.8kbps) connections. Any modern PC, MAC, or UNIX workstation can access the TerraServer using a standard web browser – Internet Explorer 3.0 or later, or Netscape Navigator 3.0 or later. If you have never used it, look at the TerraServer web site at <http://terraserver.microsoft.com/>.

There are four methods by which a user locates an image:

1. **Coverage Map:** clicking on low resolution map of the world shaded to show where coverage exists,
2. **Place Query:** entering a place name, e.g. San Francisco,
3. **Coordinate Query:** entering the longitude and latitude of interest, or
4. **Famous Place:** selecting a location from a pre-compiled list of places.

A successful search presents the user with a web page containing an HTML table of image tiles fetched from a SQL database. GIF images surrounding the image tile table provide the user with the following basic controls:

5. Pan and zoom
6. Display image meta data
7. Download an image copy
8. Control the size of the image table. There are three fixed sizes – Small (400 x 200 pixels), Medium (600 x 400 pixels), and Large (800 x 600 pixels).
9. Choose the image “style” or theme. TerraServer stores three image styles -- imagery stored in TerraServer – shaded relief, topographic map, and photograph (aerial or satellite).

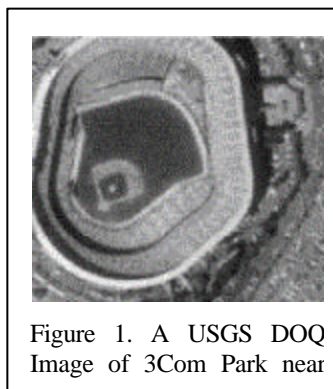
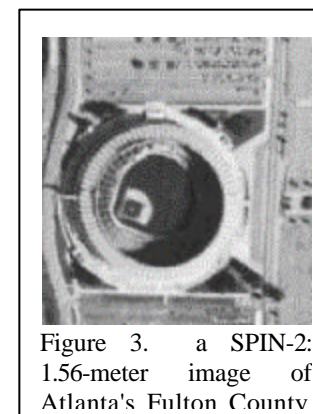
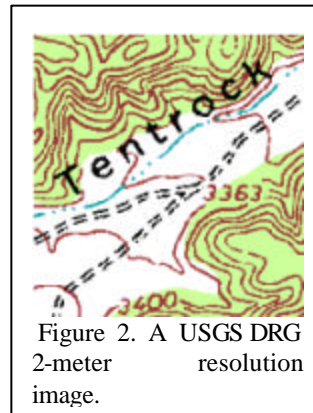
We expect and support the use of TerraServer image tiles on remote web sites. Most data on TerraServer is public domain data. Therefore, we deliberately chose simple graphics and storage methods so that users could craft their own web pages that display TerraServer image tiles.

Imagery is categorized into “themes” by data source, projection system, and image “style”. Currently, there are four data themes:

**USGS Digital Ortho-Quadrangles (DOQ)** are gray-scale or color infrared, 1-meter resolution aerial photos. Cars can be seen, but 1-meter resolution is too coarse to show people. Imagery is ortho-rectified to 1-meter square pixels. Approximately 50% of the U.S. has been digitized. The conterminous U.S. is expected to be completed by the end of 2001. Some locations have more than one DOQ image available varying by image source date or color mode. TerraServer stores the latest grayscale image. If only a color infrared image is available, they it is converted to grayscale before tiling and storing in the database.

**USGS Digital Raster Graphics (DRG)** DRGs are the digitized versions of the popular USGS topographic maps. The complete set of USGS topographic maps have been scanned for the conterminous United States and Hawaii. The original images are available in three map scales – 24,000:1 (2.4 meters/pixel), 100,000:1 (10 meters per pixel) and 250,000 meters per pixel. The raster images are re-sampled to nearest power of 2 meters per pixel.

**Aerial Images SPIN-2™** are grayscale 1.56-meter resolution de-classified Russian military satellite images. The images are re-sampled to 2-meter resolution. TerraServer contains SPIN-2 images of Western Europe, the United States, and the Far East. Unfortunately, there is little coverage of Canada, South America, Africa, and Southeast Asia. The SPIN-2 imagery is rectified, i.e. rotated so north is up, but is not ortho-rectified. That is, the image is not “rubber sheeted” so that each pixel covers a consistent square number of square meters.



However, given the height of the satellite, the difference in ground area between individual pixels is small.

**Encarta Shaded Relief** is natural color, shaded relief map of the globe. The full resolution image detail is approximately 1 kilometer per pixel. The image is a seamless image of the globe between latitude +80° and -80°. The Microsoft Geography Business Unit assembled the image from a public domain combination of weather satellite data and elevation data. The image appears in the Encarta Virtual Globe add-on product to the Encarta Encyclopedia CD title.

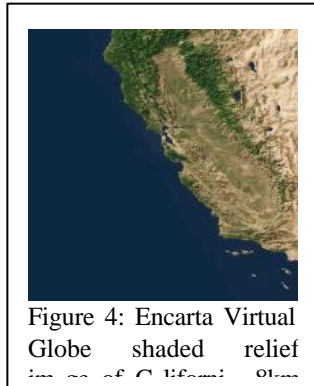


Figure 4: Encarta Virtual Globe shaded relief image of California.

## 2.1. System Architecture

TerraServer is a “thin-client / fat-server” design. The TerraServer has a 3-tier architecture:

**Tier 1: The Client** is a graphical web browser or other hardware/software system that supports HTTP 1.1 protocols and HTML 3.2 document structure. TerraServer is built and tested with Netscape Navigator and Internet Explorer on Windows, MacOS, and UNIX.

**Tier 2: The Application Logic** is a web server application that responds to HTTP requests submitted by clients by interacting with the Tier 3 database system and applying logic to the results returned.

**Tier 3: The Database System** is a SQL Server 7.0 Relational DBMS containing **all image** and meta-data required by the Application Logic tier.

Most web pages are dynamically formed and contain standard HTML 3.2 tags. All imagery is stored within SQL Server “blob fields” and compressed in Jpeg or Gif format. There is one row per image tile. The row contains the meta-data describing the tile’s geographic attributes and one blob field containing the image data. Imagery is presented via <IMG SRC=...> tags without the aid of java applets, or other specialized client side controls. The SRC URL references a script executed on the web server that is fetched from the database and sent back to the browser prefixed with the appropriate mime type. TerraServer supports Netscape Navigator and Internet Explorer V3.0 or later browsers.

“Zoomed out” or sub-sampled imagery is also stored in the database, one-row per tile. Four higher resolution tiles are sub-sampled into one lower resolution tile. The process is repeated for the number of levels in the image hierarchy. We found, and our graphics colleagues have confirmed, that a 7 level image pyramid is the maximum for grayscale imagery. All levels of the image pyramid are pre-

computed and stored in the database for the following reasons:

1. We wanted to build the largest physical database that we could.
2. A 7 level image pyramid would require 25,600 tiles to be sub-sampled to create one single 64:1 resolution tile. We do not believe users are willing to wait for this operation to be completed “on-the-fly”.
3. We did not have the resources to develop and support a high performance, server-side sub-sampling and dynamic image generation application.<sup>1</sup>
4. We wanted users to reference TerraServer imagery on their own web pages with a simple <IMG SRC...> tag and not require a client-side control to display and sub-sample the imagery.

The web site is a cluster of machines. A set of servers executes Microsoft Internet Information Server (IIS) web server software that interfaces with the SQL Server databases. The site is designed to support a variable number of web servers for performance (more net cards for increased bandwidth) and reliability reasons (a failed web server does not take down the whole web site). Increasing the throughput of the web site is as simple as adding another web server until the network web and database servers saturate.

The web servers connect to the database servers that host the SQL Server database via a separate internal network. This protects the SQL Servers provides an extra level of security from hackers and a separate private network that does not compete with Internet or other network traffic. Currently, there are two database servers – one for aerial and satellite imagery, and a second for the topographic maps. There are also two 100mbit subnets between the database and web servers.

The number and size of database servers is determined by the popularity of the data and convenience. The bandwidth between database servers and web servers drives the hardware configuration. Topo maps are expected to be popular, so we decided to host them on a separate machine where queries for aerial photography data do not have to compete for bandwidth with queries for topo data. The shaded relief data is small and replicated on both the topo and aerial photography server.

Web pages containing imagery have a consistent layout. Users can control the number of image tiles that appear on a single page. The user’s monitor size and Internet

---

<sup>1</sup> The entire TerraServer web and database application was developed by one person.



connection speed dictate their choice. Web pages are dynamically created on the web servers due to the millions of combinations of possible web pages.

There are a wide number of choices for dynamic web page construction on Internet Information Server based web servers – ISAPI, Active Server Pages, CGI/Perl, Cold Fusion, etc. We chose Active Server Pages (ASP) for a number of reasons:

- ?? Fast and easy development – ASP host Visual Basic or JavaScript engines. An ASP document can be written and debugged without requiring a compiler or other elaborate development tools.
- ?? The execution time of our ASP scripts was dominated by the SQL execution time and the data transfer time between database and web server. There was little or no performance gain in using a compiled language.<sup>2</sup>

We chose Visual Basic as the scripting language because it had better support for error handling. The Active Data Object (ADO), an OLEDB object, is used to access the SQL Server database engine. The Visual Basic error object could trap the errors raised by the ADO object. Our Visual Basic scripts process URL query strings, access the Imagery SQL database, and dynamically generate HTML pages. One ASP page, Tile.asp, is responsible for retrieving Jpeg or GIF formatted blobs from the database and sending it back to the client. A second ASP page, image.asp, is responsible for executing SQL queries to fetch the meta data necessary to dynamically produce the HTML tags which format an image page. These two web scripts are called 85% of the time.

The Cmap.asp (coverage map), Place.asp, Geo.asp, and Famous.asp implement the four search methods described previously. Each major function, e.g. Download, Image Information, the Home Page, etc., is implemented by a separate ASP page.

All TerraServer ASP scripts have a common structure. Database access is performed by calling a single SQL Server stored procedure function. The SQL stored procedure returns one or more record sets. The ASP script calls ADO methods to connect to the database server, call one stored procedure, iterate through the returned result set(s), and disconnect. This design achieves the highest performance as we learned during the first few weeks of service.

---

<sup>2</sup> Originally we built the web application in C and accessed the web server via the ISAPI interface. We abandoned this approach after determining there was no performance gain in our case and a substantial increase in development cost.

## 2.1 Tuning the Application

TerraServer was our group's very first web site. While we had some professional graphics design assistance, we developed the web application by the classic seat-of-the-pants method. Also like most software projects, particularly Internet projects, we were under marketing pressure to release to the web quickly. We learned a lot about our design and products we chose during an all too brief beta period and during the first month of live service.

We initially estimated the application was interesting enough to generate 1 million hits or 250,000 page views a day. Later we increased our estimate to 5 million hits and 1 million page views a day. We configured 4 web servers to support the 5 million hits and 1 million page views per day estimate. Officially, TerraServer went live on June 24, 1998. However, there was an article published on the front of the USA Today Business page on June 22, 1998. The article proved we grossly underestimated the popularity of the web site.

Starting on June 22, our four web servers managed to deliver 35 million web hits and 8 million page views. Millions more were rejected. We quickly grew our site to 10 web servers by the weekend and learned the following:

1. Web server software is really a TP Monitor. Once we realized this point, we used the tuning skills we learned back in the late 70s and 80s to good use. We treated the database server as a scarce resource and used the web server configuration tools to optimally schedule requests to the back end. Prior to this discovery, we unleashed requests from the web servers to the backend via a "fire hose" and were genuinely surprised when the database server ground to a halt.
2. Round trips to the database server are costly. Therefore, do as much as possible in one trip.
3. People look at imagery of where they live. While spending many a sleepless night the first week, we noticed that there was the interesting "sine wave" of Internet connection and disk activity. In highly populated and covered areas, we would notice a precipitous rise in user connections at the start of that time zone's day. Between 5 am and 6am PST or 8 am and 9am EST, the number of user connections would rise steeply. About one hour later, the number of connections continued to rise, but the disk activity began to drop and reach a steady state. Over time, we realized that separate users were requesting the same data as their neighbors. We had 2 GB of physical memory on the database server, about 1.8 GB was SQL Server's memory cache. Thus many of the queries were resolved out of the database cache.

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

## LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

## FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

## E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.