

INTERNET ARCHIVE  
**WayBackMachine**

http://terraserver.microsoft.com:80/terra\_story\_interface.asp

Go

2 captures  
7 Jul 2000 - 1 Oct 2000

JUN JUL OCT  
1999 07 2000 2001

f

About this capture

**msn.**

The foundation for Microsoft Windows 2000?

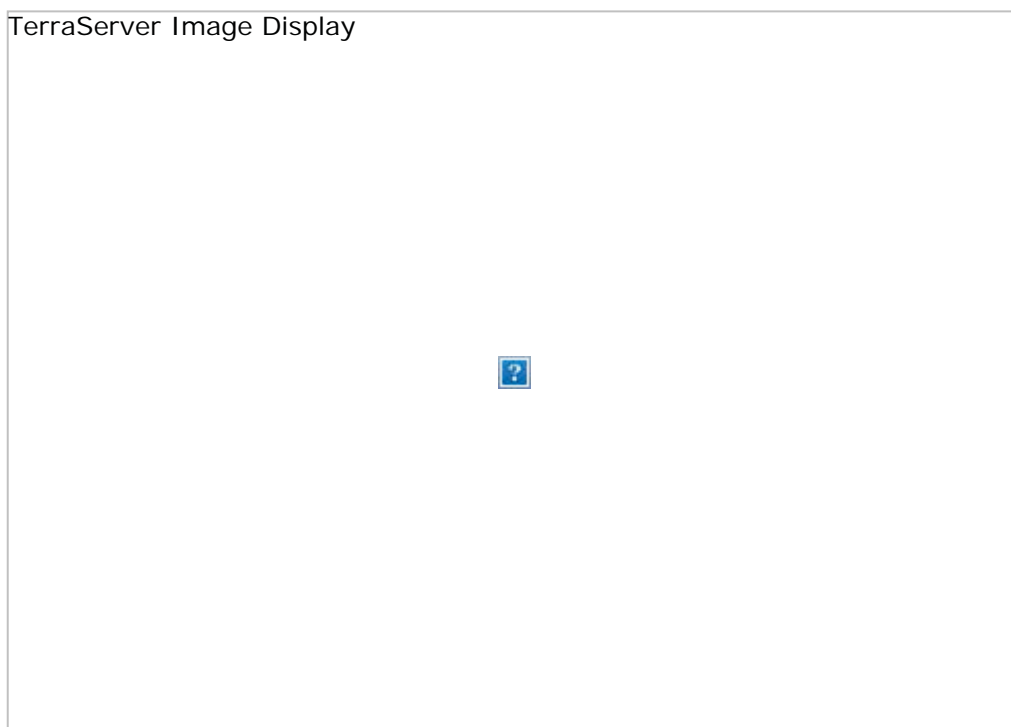
**COMPAQ**

**Encarta Learning Zone** > TerraServer

[Reference](#) [Homework](#) [Parents](#)  
[College](#) [Teachers](#) [Features](#)

[TerraServer](#) Find  [go](#) [Advanced Find](#) [About](#) [Famous Places](#)

## TerraServer Interface



In the previous sections of the TerraServer Story, we discussed the loading and editing process TerraServer goes through to get all its imagery on-line. In this section, we will explore what happens when you want to browse imagery on TerraServer.

Microsoft TerraServer can be accessed from any web browser. The image display web page is a simple HTML document containing a table of image source tags identifying the specific image tiles to form the picture. Anchor tags, or buttons, allow you to pan and zoom through the tiles of a single theme at a time. Any web browser that supports HTML tables and display of JPEG and GIF images can host the TerraServer user interface.

TerraServer uses standard HTML whenever possible. However, we must use Java applets on some of our web pages because they enable us to perform specific functions that are not possible to do in HTML. We use the applets for such functions as Street Map, downloading images, and viewing full resolution SPIN-2 images.

The imagery and web application is designed to be easily accessed by automated processes through HTTP protocols. TerraServer uses Active Server Page (ASP) scripts, or Web interfaces, to send complete HTML documents to web browsers connected over the Internet. The web browser decodes the HTML and formats the user's browser window. The web browser opens the image tile URL that causes the web

TerraServer is built and tested with Netscape Navigator V3.0, V4.0, and later, and Internet Explorer V3.0, V4.0, and later. However, any web browser that supports HTML 3.2, JPEG images, GIF images, and Java applets can interact with TerraServer. Between 65 and 75 different browsers based on platform, browser type, and browser version visit TerraServer every day.

The TerraServer web site is configured to support a maximum of 6,000 simultaneous web browser connections.

The Microsoft Internet Information Server standard software implements an HTTP 1.1 compliant web server. It handles all requests received from web browsers and dispatches each request to the appropriate server component. Requests for HTML files are transferred directly to the web browser without any interaction or intervention by TerraServer application code.

When you click "Go" on the Find a Specific Place and Advanced Search pages, an ASP script is invoked to perform the query by calling upon the appropriate TerraServer application module. There are 23 separate ASP scripts that implement the entire TerraServer web application.

There are five separate Active Server Page modules that implement the TerraServer search system and provide several methods to navigate to an initial spot on the globe where a user can pan and zoom an image:

- Search-By-Specific-Place (*place.asp*) - Searches the Gazetteer tables in the TerraServer SQL database for matching place names.
- Search-By-Address (*address.asp*) - Searches the TerraServer SQL database for matching place locations.
- Search-By-Coordinate (*image.asp*) - Locates the TerraServer image tile that overlaps the specified latitude and longitude point.
- Click-On-Coverage-Map (*cmap.asp*) - Translates a click on the shaded coverage map to a latitude and longitude and then does a Search-By-Coordinate operation.
- List-Famous-Places (*famous.asp*) - Retrieves listed place descriptions and known URLs to TerraServer image pages.

The Search-By-Specific-Place and List-Famous-Places functions return an HTML page containing a list of imagery indexed by place or descriptive name. The user can click on a URL link that will invoke a TerraServer image display ASP page. The Search-By-Coordinate and Click-On-Coverage-Map page directly call the image display ASP page.

TerraServer ASP image display page, *image.asp*, decodes its parameters from the calling URL's query string. Parameters are the key fields to locate a TerraServer image in the database:

- Scale - Resolution of the image to display.
- Theme - Data provider and image type to display.
- X - Relative integer position along the X-Axis of the image within the seamless entity.
- Y - Relative integer position along the Y-Axis of the image within the seamless entity.
- Z - Seamless entity identifier.
- Window Size - Small, Medium, or Large.

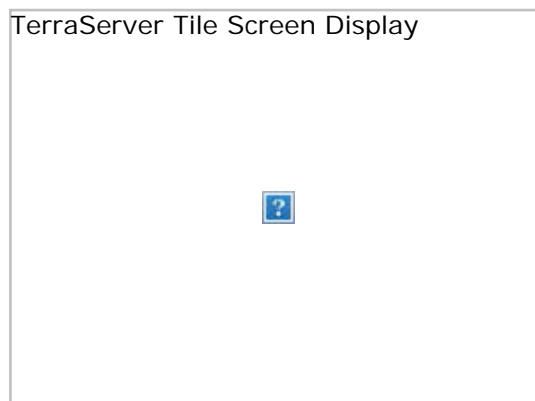
These fields identify the table and row containing the database image to display in the center of the screen. The Window Size parameter is used by the procedure to compute the number of surrounding image tiles to display. Buttons are provided to control the number of columns and rows in the table. The user can select Small (2 columns by 1 row), Medium (3 columns by 2 rows), and Large (4 columns by 3 rows).

The size of the tiles (200 by 200 pixels) and image table sizes were selected after experimenting with various monitor resolutions and measuring download times. The tiles size was selected to compress to 10KB or less per tile and fit the following monitor resolutions:

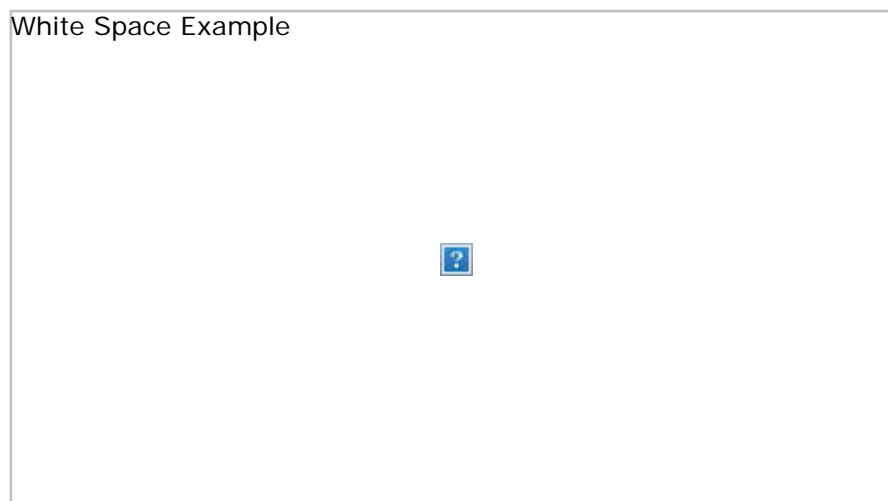
<b>Monitor Size</b>	800 x 600	1024 x 768	12 x 1024
<b>Table Size (in Tiles)</b>	2 x 1	3 x 2	4 x 3

The image display script calls a Stored Procedure to determine the presence of the center image, its coordinates, nearest place name, theme, and the presence of surrounding images. This information is used to format an HTML table containing references to TerraServer image retrieval ASP scripts.

The diagram below depicts which surrounding tiles are displayed on the page depending on the screen size. All shaded cells below identify the tiles making up a Large image. The six center cells (gray and black shaded) identify tiles forming a Medium Image. The black shaded tiles form a Small image. The open table cells identify where the image display script inserts references to pan and zoom buttons.



The image display ASP script determines the presence of each tile forming the mosaic. If a tile does not exist, the script generates a URL to a "white-noise" image. The white-noise image is intended to indicate the end of an image scene. The image below is an example of an image with white-noise. By design, it is intended to look like a matte placed around framed photography and artwork.



*Image with Example of White Space*

The script uses the image presence information to determine if the associated direction button should be enabled. For example, if the northwest corner tile is not present, the image display page does not create an active link to the "Go Northwest" button.

The HTML image source tag, `<IMG SRC="URL...">` is used to instruct the web browser how to fetch an image tile from the TerraServer database. The image display page builds the URL to the image fetch ASP page, `tile.asp`, using the input Scale, Theme, X, Y, and Z parameters passed to the image display page as a starting point.

The `tile.asp` script decodes the parameters found in the URL's query string and builds a call to a TerraServer SQL Stored Procedure. The SQL Stored Procedure locates the image record in the appropriate

image cannot be located, then the row found in the **NoImage** database table is returned. This ensures that an image data blob is always returned to the *tile.asp* script.

The *tile.asp* scripts writes the image blob back to the web browser. The image blob is a fully formed Jpeg or GIF file as described by the mime/type string. The browser decompresses the image file and displays it in the appropriate spot in the image mosaic formed by the HTML table.

All pan and zoom buttons are references to the *image.asp* tag. The Search pages usually start the user at the 16-meter resolution level. Most users pan, click on the image to zoom to 8-meter resolution, pan some more, click on the image to zoom to 4-meter, etc. before they invoke another search page. Thus, the *image.asp* and *tile.asp* pages are the workhorses of the Microsoft TerraServer system and account for well over 90% of web hits. The five search pages, *image.asp*, and *tile.asp* account for over 98% of all web hits on TerraServer.

The Internet Information Server software is tuned to reflect the fact that seven ASP pages perform the majority of the workload. The scripts are compiled and pre-buffered so they are always available for immediate execution.

All meta and image data is stored in a single Microsoft SQL Server 7.0 database. Multiple database servers can be configured into a single TerraServer web site. The only rule is that all data for a theme must be contained entirely in one database. A theme's data can be replicated to one or more backup servers, but a theme's inventory cannot be split across two database servers.

All end user access to TerraServer image and meta data occurs through TerraServer SQL Stored Procedures. The Application Logic web servers access the SQL Stored Procedures via an Active Data Object (ADO) interface that is layered on top of the ODBC API.

TerraServer's unique processes and interface allow users access to imagery that was once only available to a select few and which can now be viewed by anyone without fees imposed by Microsoft, TerraServer partners, or the data providers. Through TerraServer, terabytes of imagery data is now available to everyone who has access to the Internet.



Microsoft  
**Research**  
 TerraServer Built by  
 Microsoft Research



#### Other Links:

[Free Games](#)  
[White Pages](#)  
[Greeting Cards](#)  
[Buy Books](#)  
[Event Tickets](#)  
[More...](#)

#### Special Features:

[Free tax filing](#)  
[Faster connections & fewer busy signals](#)  
[Free credit analysis](#)  
[Enter to win a \\$1,000 shopping spree](#)  
[For women only \(no men allowed\)...](#)  
[More...](#)