

# GISInternals Tools and Resources

Tamas Szekeres

MapServer, GDAL, QGIS developer at OSGeo, szekerest@gmail.com

**Abstract:** The open source software development methodology allows the direct link between the developers and users. The new feature requests, the bug reports or patches can be submitted directly to the project issue trackers and the developers can review and apply the changes in a relatively small delay. Despite of this flexibility in the development, we often find the projects cumbersome in terms of "software delivery", the new features or fixes in the codebase may not appear in the binary distributions in time, or the project itself may not be "ready to use" in the same way as what a proprietary software can usually provide. This is especially the case with the softwares located in the middle tier like GDAL or MapServer which are probably deployed as the part of a larger system in most cases.

## Introduction

Regarding to the open source geospatial projects at OSGeo, there are only a few binary distributions available for Windows platforms and each of them have specific advantages or disadvantages. Although some of the projects may provide official binaries (like for the desktop applications as QGIS and GRASS) many of them only have the source code available "officially" on the download page.

The official binaries provided by the installers may be bound to the latest release versions and only few of them provide daily built binaries which would be essential to deliver that latest "bleeding edge" technologies to the end user.

## Build system

GISInternals is an online system for creating daily built binary packages for the GDAL and MapServer projects in a non profit fashion [1]. The ancestor of this system has been provided to be the Windows buildslaves for the OSGeo buildbot in 2007. The build system in the current form (providing downloadable packages) has been set up in 2009. As of this time, the system has been continuously improved by adding more and more packages to make the life easier for the users and developers of these open source projects. During this time the number of the visitors and the amount of the downloads continues to grow, and the site has been visited from 195 countries around the world providing almost a complete geographical coverage.

Automated compilation of the projects is scheduled daily at 4:00AM (Central European Time) and the complete build process takes about 4 hours. The total size of all the packages created daily is at about 1 gigabytes.

This build environment compiles GDAL and MapServer from the source repositories (git and svn) and provide cutting edge binary packages containing the latest fixes until the time when build has started (refer to build status for the details). Only the binaries of the latest successful builds are packaged and made available to download. The possible breaking changes in the repositories cause the compilation to terminate immediately before the packaging is started.

The primary objectives of this build system are:

- Provide complete solutions (containing all the dependencies for running and compiling the applications).
- Provide homogenous builds. All the dependencies are compiled against the same CRT libraries, to eliminate the potential run-time problems.
- Downloadable packages are kept as small as possible (only the required files are included).
- Packages are prepared to have the things set up and running easily.
- Packages are location independent. You can run and develop multiple versions of the projects in the same computer in different directories.

In addition to the binary packages and installers, the build system provides all the required files (libs and headers of all dependencies) to compile the projects on Windows by using the supported Microsoft Visual Studio versions. The automated build process itself is also based on these SDKs and each package contains a master Makefile that can be used to compile MapServer and GDAL easily from a Visual Studio Command Prompt.

## **Package versions**

Packages are being built in parallel using several MSVC compiler versions. At the moment MSVC 2005, 2008 and 2010 versions are supported for both Win32 and Win64 target architectures. We also have plans to add packages for MSVC 2012+ version in the near future.

The binaries from the different packages cannot be mixed or interchanged, but existing distributions created by using the same compiler versions or target architectures may be upgraded with these files.

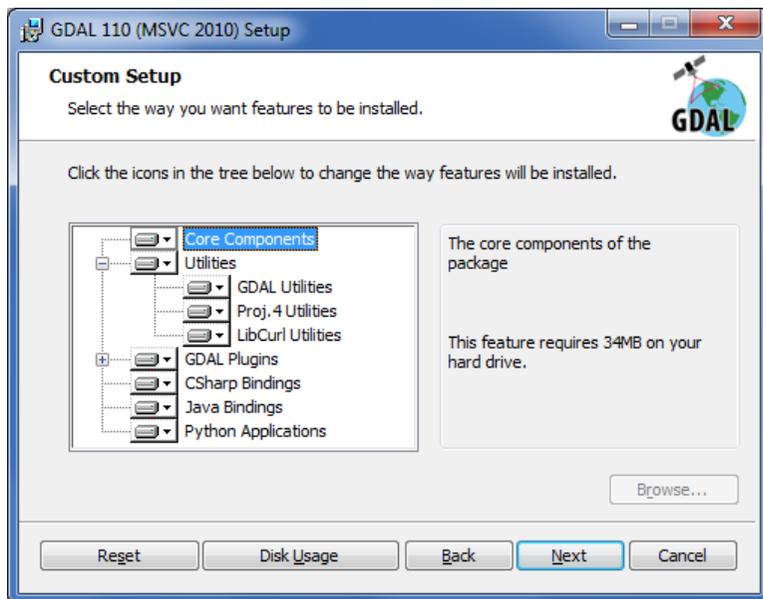
Multiple versions of the packages can be installed side by side into different directories on the same machine.

## Command line tools

Either the zip packages or the core msi installers provide options for running a preconfigured command prompt from which the console applications (provided by GDAL and MapServer) can easily be executed. For more information about the utility programs of GDAL and MapServer please refer to [2], [3] and [4]

## GDAL installers

The build system provides daily built msi installers for all package versions of GDAL/OGR (*Figure 1*). The core installer deploys a shortcut to a preconfigured command prompt, where all GDAL tools (like ogr2ogr, gdal\_translate etc.) are available to be executed directly. Some of the plugins - having custom licenses (ECW, FileGDB, Oracle, MrSID) - are available from separate installers. Generic msi installers for the python bindings from Python v2.7 to v3.3 are also available for each package version.



*Figure 1. GDAL installer (feature selection page)*

## MapServer installers

GISInternals provides a separate installer to set up the CGI MapServer in IIS, which is a quick and easy way to try the latest MapServer builds in action (*Figure 2*). The existing installers (like ms4w or OSGeo4w) have bundled Apache versions for this purpose, but it might also be reasonable to use

the existing IIS installation for testing or even for production on Windows platforms. Upgrading to a new version may be as simple as running the installer again or just replacing the binary files from a compatible package.

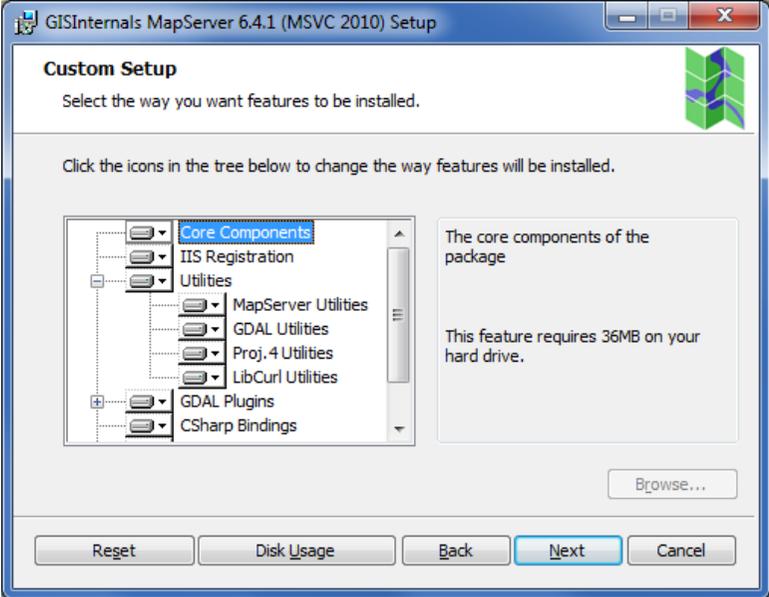


Figure 2. MapServer installer (feature selection page)

## MapManager

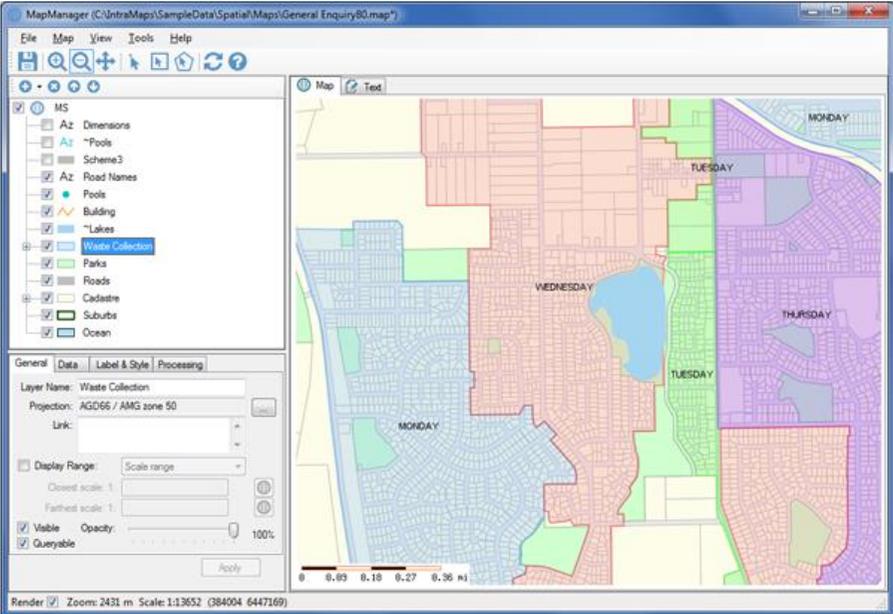


Figure 3. MapServer MapManager user interface

Since MapServer doesn't provide an "official" GUI to create or modify the map configurations (map files), it's quite difficult to achieve the expected look by using a text editor when configuring the layers and styles. MapServer MapManager was aimed to bridge this barrier and allow the users to utilize MapServer in their GIS based systems easier by using an intuitive graphical user interface (*Figure 3*).

MapManger is provided as daily built msi installers based on the latest release and stable branches of MapServer, the complete description of the software is available online [5].

MapManager provides a feature rich user intrface to configure map files by supporting -among others - the following options.

*Table 1. Key feaures of MapServer MapManager*

|                                             |                                                                                                                                                                                                                                                                                      |
|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Map display with zooming and panning        | The map control provides the option to zoom and pan using the mouse                                                                                                                                                                                                                  |
| Feature selection                           | Provides feature selection by point, rectangle and polygon shapes                                                                                                                                                                                                                    |
| Tree view for MapServer objects             | Displays the hierarchy of the map, layer, class, style and label objects                                                                                                                                                                                                             |
| Style library                               | The style library provides a common repository of the styles which can be applied to the layer styles easily                                                                                                                                                                         |
| Tile Manager                                | The tile Manager can be used to create a set of tiles to be served by MapServer as a TMS. Tile Manager allows you to design a tile set from a MapManager map file with the option to set how many levels, image format and desired machine resources to use while creating the tiles |
| Theme Wizards with color ranges             | The Theme Wizards provide an easy to use interface to define a thematic symbology for a layer quickly. The Theme Wizards can be invoked by right clicking on a layer and selecting the Add Theme context menu in the Layer Panel                                                     |
| Options for adding vector and raster layers | MapManager provides dialogs for adding file based vector/raster layers, adding WMS layers or adding layers from MSSQL spatial databases                                                                                                                                              |
| Projection selection dialog                 | Custom dialog is provided for finding and assigning projections for the complete set of projections supported by Proj.4                                                                                                                                                              |
| Save map image in various formats           | The generated map image can be saved to various output formats (jpg, png, gif, pdf, svgm GTiff, kml, kmz) provided by MapSever                                                                                                                                                       |
| MapManager Command Prompt                   | A shortcut for a command prompt is installed in the start menu for running the GDAL and MapServer console applications                                                                                                                                                               |

## OSGeo4w packages

The build system at GISInternals provides daily builds for MapServer to OSGeo4w which are automatically uploaded to the OSGeo4w repository.

In the package selection list of the OSGeo4w installer the installation can easily be upgraded to the latest changes have been applied in the MapServer codebase. (Figure 4).

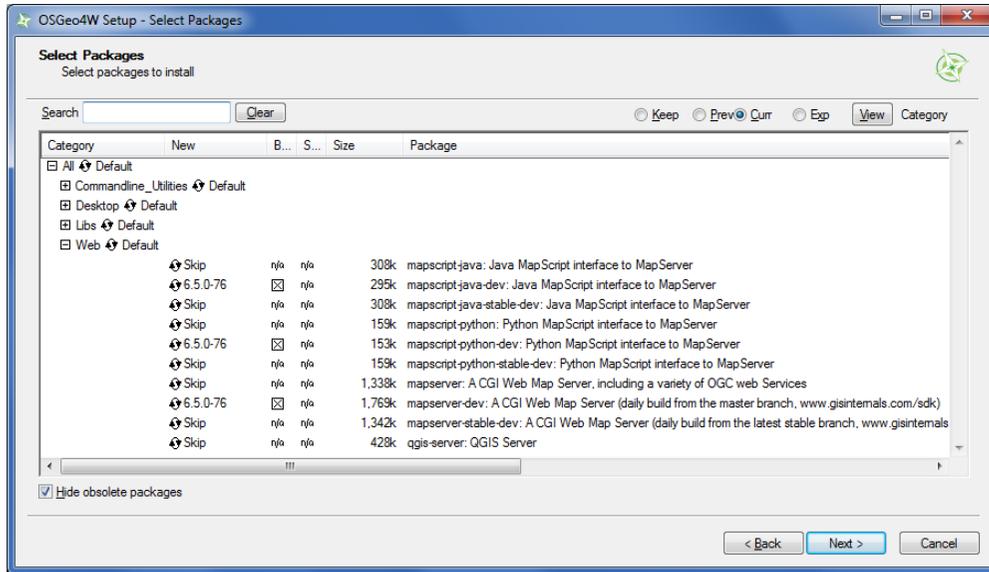


Figure 4. MapServer packages for OSGeo4w

The daily built version of MapServer provided for OSGeo4w are based on the latest stable (mapserver-dev) and the master branches (mapserver-stable-dev)

## Build support for MapWinGIS

One of the primary objectives of the build system is to provide software development kits (SDKs) for the developers creating GDAL and MapServer based applications. The provided build kits are very small in size, there's no specific requirements about the installation, the packages can be copied to any locations and multiple versions could also be installed side by side. The most common practice is to use a separate copy of the SDK directory for each separate projects which corresponds to the actual Visual Studio version (ie VS2008 for MapWinGIS). Each target architecture (x86 or x64) requires a separate SDK so the project configurations should switch the library locations according to the actual settings.

When building MapWinGIS the release-1500 and release-1500-x64 versions should be downloaded and copied into the proper locations within the SupportingLibraries subdirectory. For more information please refer to the instruction (Download files.txt) in the MapWinGIS source code repository [6].

## Terms of use

The packages from GISInternals are provided with a fully permissive license [7]. It is intended to give you permission to do whatever you want with the files: download, modify, redistribute as you please, including building proprietary commercial software, no additional permission from the binary distribution provider is required. Some external libraries which can be optionally used by GDAL and MapServer (provided as plugins) are under radically different licenses, and valid licenses might be obtained separately for each of these dependent libraries.

## Issue tracking

A github issue tracker has been set up for the build system to receive the feature requests or bug reports from the users [8].

## Online references

[1]: GISInternals support site

<http://www.gisinternals.com/sdk/>

[2]: GDAL Utility Programs

[http://www.gdal.org/gdal\\_utilities.html](http://www.gdal.org/gdal_utilities.html)

[3]: OGR Utility Programs

[http://www.gdal.org/ogr\\_utilities.html](http://www.gdal.org/ogr_utilities.html)

[4]: MapServer Utilities

<http://mapserver.org/documentation.html#utilities>

[5]: MapServer MapManager Online Help

<http://www.gisinternals.com/MapManager/>

[6]: MapWinGIS source code repository

<http://mapwingis.codeplex.com/SourceControl/latest>

[7]: GISInternals license terms

<http://www.gisinternals.com/sdk/license.txt>

[8]: Issue tracker for the GISInternals build system

<https://github.com/gisinternals/buildsystem>