Installing GD Graphics Library
on
Mac OS X Server 10.5.x
for 64- and 32-bit CPUs

1. - Introduction

NOTE: With the introduction of Mac OS X Server 10.5 Leopard a few things have changed with regards to included libraries and PHP5. For example, the mysql client libraries are not present anymore, so recompiling PHP exactly as intended by Apple is not possible. It still is possible to update PHP5 when new versions come out, but it will require a few workarounds. Given this situation, we have decided to split the GD and PHP tutorials into 10.4.x and 10.5.x versions. Furthermore, the 10.5 tutorial on the GD Graphics Library is now available in two flavors. One purely PHP centric and one universally applicable, like the one you are reading.

The purpose of this document is to provide instructions on how to install the GD Graphics Library on OS X 10.5.x Server. Since there are certain differences, it has separate instructions for 64- and 32-bit CPUs.

You will not find many explanations as to why something is done one way or the other. There are plenty of tutorials on this around on the internet. This document should just help you in getting things done.

The GD Graphics Library depends on other libraries to be installed first. These include: zlib, libpng, libjpeg et al. Luckily, the Developer Tools for OS X Server 10.5.x include current versions for all but libjpeg.

This document will require you to use the command line. If you do not feel comfortable with using the command line, you should look for a ready made installer package or for somebody to assist you.

This document is written for Mac OS X 10.5.x. There is another document available for Mac OS X 10.4.x

DISCLAIMER: Whatever you do based on this document, you do it at your own risk! Just in case you haven't understood: Whatever you do based on this document, you do it at your own risk!

This tutorial has been tested on a standard Mac OS X 10.5.x Server installation. If you have already tinkered with your

system, be aware that things might differ. It is impossible for me to foresee all changes that one might have applied to a server.

This tutorial contains step-by-step instructions for the terminal. Although you could just type them in line by line, it is recommended you have a basic understanding of the terminal.

NOTE: Throughout this document you will see a few instructions that differ between 32- and 64-bit CPUs. Leopard Server is built with Universal Binaries, which contain 4 architectures (32- and 64-bit for PPC and Intel). On a 64-bit system, Apache2 will try to load the 64-bit image of libphp5.so, so it is important that Apache2 can find it. Thus, we must compile PHP5 and all related modules as the same type of Universal Binary. Well, you could also just build versions relevant to your system, but I am not going to publish x variations ;-)

(This is a very simplified version of what really goes on, just so you understand certain apparent redundancies).

2. - Requirements

Before you get started, you need to make sure some basic requirements are met:

- You have made a backup of your system.
- You have the latest version of Apple's Developer Tools (XCode 3.0 or higher for 10.5.x) installed.
  Dev Tools are available on your Server DVD and as a free download from Apple's Developer Connection.
- X11 is installed (part of standard OS X Server install unless you deselect it)
- X11 SDK is installed (available on your OS X Developer Tools Disc or Image and part of the standard installation. This is different from the X11 client that comes with OS X.)
- You do have a backup
- You are running 10.5.x
- You have not manually updated anything related to GD, libpng, libjpeg and freetype so far (if you have, you must know how to adapt these instructions to the changes you made).

3. - Getting and installing libjpeg

This chapter will guide you through installing libjpeg. It is required for GD to work.

So let's get going:
Make sure you are logged in as root.

Get and install the latest version of the libraries by issuing the following commands (*in oblique type*). Issue them one after the other making sure you do not miss any dots or slashes. Also note that the download URLs may change in the future. In that case just replace the URLs in this document with the current one. Lines wrapping without line spacing are a single command.

(First we create our environment)

mkdir -p /SourceCache

cd /SourceCache

(Now we get the libraries and unpack them.)

curl -O http://www.ijg.org/files/jpegsrc.v6b.tar.gz

curl -O http://www.libgd.org/releases/gd-2.0.35.tar.gz

tar xzpf jpegsrc.v6b.tar.gz

tar xzpf gd-2.0.35.tar.gz

(Now we install libjpeg)

cd /SourceCache/jpeg-6b

cp /usr/share/libtool/config.sub .

cp /usr/share/libtool/config.guess .

NOTE: The following configure command is different for 32- and 64-bit CPUs

NOTE: Just in case you missed it: The following configure command is different for 32- and 64-bit CPUs

For 32-bit use:

./configure --enable-shared

For 64-bit use:

MACOSX_DEPLOYMENT_TARGET=10.5 CFLAGS="-arch ppc -arch ppc64 -arch i386 -arch x86_64 -g -Os -pipe -no-cpp-precomp" CCFLAGS="-arch ppc -arch ppc64 -arch i386 -arch x86_64 -g -Os -pipe" CXXFLAGS="-arch ppc -arch ppc64 -arch i386 -arch x86_64 -g -Os -pipe" LDFLAGS="-arch ppc -arch ppc64 -arch i386 -arch x86_64 -bind_at_load" ./configure --enable-shared

make

mkdir -p /usr/local/include

mkdir -p /usr/local/bin

mkdir -p /usr/local/lib

mkdir -p /usr/local/man/man1

make install

4. – Installing the GD Graphics Library

This chapter will guide you through installing the GD Graphics Library.

So let's get going: Make sure you are logged in as root.

Install the latest version of GD by issuing the following commands (*in oblique type*). Issue them one after the other making sure you do not miss any dots or slashes. Lines wrapping without line spacing are a single command.

mkdir -p /SourceCache

cd /SourceCache/gd-2.0.35

NOTE: The following configure command is different for 32- and 64-bit CPUs

NOTE: Just in case you missed it: The following configure command is different for 32- and 64-bit CPUs

For 32-bit use:

./configure

For 64-bit use:

MACOSX_DEPLOYMENT_TARGET=10.5 CFLAGS="-arch ppc -arch ppc64 -arch i386 -arch x86_64 -g -Os -pipe -no-cpp-precomp" CCFLAGS="-arch ppc -arch ppc64 -arch i386 -arch x86_64 -g -Os -pipe" CXXFLAGS="-arch ppc -arch ppc64 -arch i386 -arch x86_64 -g -Os -pipe" LDFLAGS="-arch ppc -arch ppc64 -arch i386 -arch x86_64 -bind_at_load" ./configure --disable-dependency-tracking

make

make install

You are now all set and should have the GD and supporting libraries installed.

For most users the next step would be to integrate the GD Graphics Library into PHP. A separate tutorial on re-building PHP (with GD support) on OS X 10.5.x Server is available on http://osx.topicdesk.com/.

5. - Caveats

Not that many. The most important caveat is that if building one
library goes wrong, the rest of the succession will most likely
not work either.

If you have tried to install one of the libraries in the past,
you may have to remove and re-install them (unless you did
install them as described).

If you have modified any paths and or environment variables,
make sure you check them against above instructions.

That's all folks.
Hope this helps.
Have fun,
Alex

Doc. Version 1.0.5, 16.11.2007
Athanasios Alexandrides
Lugano, Switzerland
tutorials -at- topicdesk.com