Building MapServer Enterprise For Linux

Contents

Building and Installing required components

Building and Installing PHP 5.0.5 from source Distribution

Build the source code.

a) The source code is available in the Oem/php directory or you can download it from www.php.net (version 5.0.5)
b) If from the Oem source you may have to change the permission setting for configure, config.* , Makefile.frag , Makefile.global etc.)
c) Installing PHP 5.0.5 to use with the web extensions **requires that Apache 2.xx be installed on your system**. Should you have an older version of Apache, you will have to update .

On the unzipped or Oem/php-5.0.5 source root directory type the following on the command line

1. Now, configure your PHP. This is where you customize your PHP with various options, like which extensions will be enabled. Do a **./configure --help** for a list of available options. In our example we'll do a simple configure with Apache 2. ***Your path to apxs may differ, in fact, the binary may even be named apxs2 on your system.***
    a. On the command like execute **whereis apxs or whereis apxs2** to find out where apxs is installed.
    b. On the command line execute **./configure --with-apxs2=/usr/local/sbin/apxs** .Note: replace the path with the path that you got with the "whereis" command in step a .

2. On the command line execute **make**

3. On the command line execute **make install**

4. If you decide to change your configure options after installation, you only need to repeat the last three steps. You only need to restart apache for the new module to take effect. A recompile of Apache is not needed.

a. Note that unless told otherwise, 'make install' will also install PEAR, various PHP tools such as phpize, install the PHP CLI, and more.

5. Setup your php.ini

a. On the command line execute **cp php.ini-dist /usr/local/lib/php.ini**

b. You may edit your .ini file to set PHP options. If you prefer having php.ini in another location, use --with-config-file-path=/some/path in step 1. If you instead choose php.ini-recommended, be certain to read the list of changes within, as they affect how PHP behaves.

6. Edit your httpd.conf to load the PHP module. The path on the right hand side of the LoadModule statement must point to the path of the PHP module on your system. The make install from above may have already added this for you, but be sure to check.

a. For PHP 5:

i. LoadModule php5_module modules/libphp5.so

7. Tell Apache to parse certain extensions as PHP. For example, let's have Apache parse the .php extension as PHP. You could have any extension(s) parse as PHP by simply adding more, with each separated by a space. We'll add .phtml to demonstrate.

a. **AddType application/x-httpd-php .php .phtml**

b. It's also common to setup the .phps extension to show highlighted PHP source, this can be done with: **AddType application/x-httpd-php-source .phps**

8. Use your normal procedure for starting the Apache server, e.g.:

/usr/local/apache2/bin/apachectl start

Please do note that the above is a subset of instructions that are available at http://us2.php.net/manual/en/install.unix.php for installing PHP with apache 2.x,x on Unix/Linux . Please refer to the website for more detailed instruction or information about the various modules available for PHP.

Building FDO 3.0

OpenSourceFDO is available in a standalone gzipped tar file. (fdo-3.0.0.tar.gz)

Download the distribution from
http://mapserverfoundation.org/htdocs/dl_mirror.php?file=fdo-3.0.0.tar.gz

Extract the source from the distribution and build the FDO according to the included instructions (**OpenSourceBuild_Readme.txt**)

The installation must be carried out with the default settings.

Once installed, check the execute permissions on the shared libraries in /usr/local/gis/fdo/3.0. Some may have been installed without execute permission. chmod a+x *.so* will correct it

Download the data for unit tests from
http://mapserverfoundation.org/htdocs/dl_mirror.php?file=fdo-3.0.0_data.tar.gz

Download the documentation from
http://mapserverfoundation.org/htdocs/dl_mirror.php?file=fdo-3.0.0_docs.tar.gz

Other Requirements

The following libraries are generally installed on the Linux environment. However, it is important that you check for them

libjpeg.so – this library should be available with your Linux distribution or you can download the source from the

- **Independent JPEG Group**
  Homepage:
  http://www.ijg.org/
  Tar/GZ:
  http://www.ijg.org/files/jpegsrc.v6b.tar.gz
  Zip:
  ftp://ftp.simtel.net/[..]pub/simtelnet/msdos/graphics/jpegsr6.zip

Building the MapServer Enterprise Distribution

MapServer Enterprise is available in a standalone gzipped tar file. (mapserverenterprise-0.9.1.tar.gz)

Download the distribution from
http://mapserverfoundation.org/htdocs/dl_mirror.php?file=mapserverenterprise-0.9.1.tar.gz

Extract the source from the distribution and proceed to the next step

Building OEM Components in the MapServer Enterprise Distribution

Move to the top level directory in the distribution

On the command line execute **./build_oem.sh**

- The default targets are all release.

Notes: dbxml-2.1.8 will report install errors when it tries to install documentation. This is a known issue and the documentation in question is not a part of this distribution. Please do ignore the errors.

Building and Installing optional components

Tomcat

Download Java Runtime 5 (JRE 5.0) from http://java.sun.com/j2se/1.5.0/download.jsp and install.
Add JAVA_HOME=/usr/java/jre1.5.0_05 to your environment.
Download Tomcat 5.5.x from http://tomcat.apache.org/download-55.cgi. Grab the .tar.gz and extract to a directory of your choice (eg. /var/www/tomcat)
Execute /var/www/tomcat/bin/startup.sh. Tomcat should start and the Tomcat welcome page should be accessible through http://locahost:8080.
Download the Apache / Tomcat connector source from http://tomcat.apache.org/download-connectors.cgi. Extract the tarball, build, and install it.

- `cd jakarta-tomcat-connectors-1.2.15-src/jk`
- `cd native`
- `./configure --with-apxs=/usr/sbin/apxsmake`
- `su -c 'make install'`

Building the MapServer Enterprise and Web Extensions

Build and install the default configuration.

Move to the top-level directory in the distribution. i.e. mapserverenterprise-0.9.1

On the command line execute **./configure** (with optional build flags)

1. the default build targets are DEBUG
2. For build options and help, execute **./configure –help**

On the command line execute **make**

1. For a install **make install**

Build and install a release configuration.

On the command line execute **make clean** from the root directory (i.e mapserverenterprise-0.9.1) if you already executed a default installation

- This step is important as the object/library names are in the same location have the same name be they debug or release builds.

On the command line execute "**./configure** --enable-optimized" (with optional build flags)

On the command line execute **make** or **make install**

To disable building and Installing the web extensions or mapserver

Move to the top-level directory in the distribution. i.e. mapserverenterprise-0.9.1

To disable the mapserver build, on the command line execute "**./configure** --enable-mapserver=no (optional flags)

To disable the web extensions build, on the command line execute "**./configure** --enable-webtier=no (optional flags)

On the command line execute **make** or **make install**

To Install in a location other then the default location

The following example illustrates installing the server and web extensions in your home directory.

On the command line execute **/configure --prefix=$HOME (optional flags)**

On the command line execute **make** or **make install**

Running the Server

If it's installed in the default location. then the **mapserver** executable and the **serverconfig.ini** file are available at

- **/usr/local/mapserverenterprise/server/bin**

Other wise at the prefix you installed the server at , the mapserver is available at

- ***${prefix}***/server/bin

The **default settings** for the **serverconfig.ini** are adequate for most users. For more information on what the settings are and what values are expected, please refer to the file. It has detailed description as to what each setting is there for.

On the command line execute the shell script **./mapserver.sh**

Configuring Web Extensions

Virtual Directory structure

The 'make install' above should create a virtual directory ready structure in

- **/usr/local/mapserverenterprise/webextensions**, or
- ***Prefix*/webextensions**

The shared libraries in ***webextensions/lib*** should be added to ld.so.conf and ldconfig should be run.

Apache

The web extension directory structure makes use of aliases to simplify the client side URLs. These changes assume that "make install" was used in the default configuration and the web extensions reside in /usr/local/mapserverenterprise/webextensions.

The following changes should be made to httpd.conf.

```
LoadModule rewrite_module modules/mod_rewrite.so

#Start PHP Config
ScriptAlias /php/ "/var/www/cgi-bin/"
Action application/x-httpd-php "/php/php_cgi.exe"
AddType application/x-httpd-php .php
#End PHP Config

ScriptAlias /MapServer/MapAgent/MapAgent.fcgi
"/usr/local/mapserverenterprise/webextensions/MapServer/MapAgent/mapage
nt"
```

```
AliasMatch ^/MapServer/MapViewerHtml/([^\?])(.*)$
"/usr/local/mapserverenterprise/webextensions/MapServer/MapViewerPhp/$1
$2"
Alias /MapServer/MapViewerHtml
"/usr/local/mapserverenterprise/webextensions/MapServer/MapViewerPhp/Ht
mlViewer.php"
AliasMatch ^/MapServer/MapViewerDwf/([^\?])(.*)$
"/usr/local/mapserverenterprise/webextensions/MapServer/MapViewerPhp/$1
$2"
Alias /MapServer/MapViewerDwf
"/usr/local/mapserverenterprise/webextensions/MapServer/MapViewerPhp/Dw
fViewer.php"

Alias /MapServer
"/usr/local/mapserverenterprise/webextensions/MapServer"
<Directory "/usr/local/mapserverenterprise/webextensions/MapServer">
  AllowOverride All
  Options All
  Order allow,deny
  Allow from all
  RewriteEngine on
  RewriteRule .* - [E=REMOTE_USER:%{HTTP:Authorization},L]
</Directory>
```

MapServer/webconfig.ini file provides configuration for the web extensions. It should already be updated by "make install". The MapServer/TempDir is used for temporary files. The apache user should be granted r/w access to TempDir. For security purposes, TempDir can be moved outside of the MapServer directory scheme so it is not directly accessible via the web.

Apache should now be configured. The FastCGI agent should be accessible. Start mapserver and hit the following URL in a browser:

http://localhost/MapServer/MapAgent/MapAgent.fcgi?OPERATION=ENUMERATERESOURCES&VERSION=1.0.0 This URL should return an XML document listing the contents of the Mapserver Enterprise repository. Administrator/admin can be used.

Php

The php_awapi.so extension must be loaded and post_max_size should be increased.

Php.ini changes

```
memory_limit = 128M
post_max_size = 128M
extension_dir = "/usr/local/mapserverenterprise/webextensions/lib"
extension=libphp_AwApi.so
```

Try running php -i from a command line shell. There should be no warnings/errors generated and the text "AwApi" should appear in the output.

Apache should now be able to run the Php extensions. Start mapserver and hit the following URL in a browser:

> http://localhost/MapServer/MapAdmin/Login.php This URL should bring up the login page for the remote administrator. The default login is Administrator/admin.

Tomcat

Tomcat is attached to Apache using mod_jk. For additional reference material, see http://tomcat.apache.org/connectors-doc/howto/quick.html

> Modify httpd.conf, or create a java.conf in /etc/httpd/conf.d as below:

```
#Tomcat Integration
#Taken from http://tomcat.apache.org/connectors-doc/howto/quick.html

# Load mod_jk module
# Update this path to match your modules location
LoadModule    jk_module  modules/mod_jk.so
# Declare the module for <IfModule directive> (remove this line on
Apache 2.x)
# AddModule     mod_jk.c
# Where to find workers.properties
# Update this path to match your conf directory location (put
workers.properties next to httpd.conf)
JkWorkersFile "/etc/httpd/conf/workers.properties"
# Where to put jk logs
# Update this path to match your logs directory location (put
mod_jk.log next to access_log)
JkLogFile     "/var/log/httpd/mod_jk.log"
# Set the jk log level [debug/error/info]
JkLogLevel    info
# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
# JkOptions indicate to send SSL KEY SIZE,
JkOptions     +ForwardKeySize +ForwardURICompat -ForwardDirectories
# JkRequestLogFormat set the request format
JkRequestLogFormat     "%w %V %T"
# Send everything for context /examples to worker named worker1 (ajp13)
JkMount  /MapServer/MapViewerJava/* worker1
JkMount  /MapServer/JavaViewerSample/* worker1
```

> Add a redirect to Tomcat in httpd.conf for the MapViewerHtml directory. Change localhost to [machineName] as appropriate.

```
# MapViewer to MapViewerJava aliases
Redirect /MapServer/MapViewerHtml
http://localhost/MapServer/MapViewerJava
```

Create a workers.properties text file in the same directory as httpd.conf with the following content:

```
# Define 1 real worker using ajp13
worker.list=worker1
# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=localhost
worker.worker1.port=8009
worker.worker1.lbfactor=50
worker.worker1.cachesize=10
worker.worker1.cache_timeout=600
worker.worker1.socket_keepalive=1
worker.worker1.recycle_timeout=300
```

Add a MapServer.xml context file to /var/www/tomcat/conf/Catalina/localhost

```
<!--

    Context configuration file for the Tomcat Manager Web App

    $Id: manager.xml,v 1.3 2004/08/26 17:03:34 remm Exp $

-->


<Context
docBase="/usr/local/mapserverenterprise/webextensions/MapServer"
         privileged="true" antiResourceLocking="false"
antiJARLocking="false">

  <!-- Link to the user database we will get roles from -->
  <ResourceLink name="users" global="UserDatabase"
                type="org.apache.catalina.UserDatabase"/>

</Context>
```

Add /usr/local/mapserverenterprise/webextensions/lib to ld.so.conf and run ldconfig.

Add JAVA_OPTS="-Djava.library.path=/usr/local/mapserverenterprise/webextensions/lib" to your environment.

Restart apache using /usr/sbin/apachectl restart.

Start Tomcat by executing /var/www/tomcat/bin/startup.sh

Check to see if Tomcat is working by hitting http://localhost:8080. You should see the Tomcat welcome page.

See if Tomcat/Apache integration is working by http://localhost/MapServer/MapViewerJava/fake.jsp. A Tomcat 404 error should be returned.

Check that java web extensions are working with http://localhost/MapServer/MapViewerJava/HtmlViewer.jsp. This should generate an "Invalid repository type" message.

Finally, verify the redirect with http://localhost/MapServer/MapViewerHtml. The "Invalid repository type" message should be returned.