



MXD2map Documentation

Release 1.2.0

Raimund Renkert, Andre Heinecke

16.11.2012 16:42

CONTENTS

1	Licence	1
2	Installation from Source on Windows	5
2.1	Requirements	5
2.2	Install Apache Webserver	5
2.3	Install UMN MapServer	6
2.4	Install the Converter	6
2.5	Compile the converter-sources	7
2.6	Setup environment-variables	7
2.7	Configure Logging	9
3	Installation from Source on GNU/Linux	10
3.1	Requirements	10
3.2	Install ArcGIS SDK	10
3.3	Install additional libraries for mxd2map	11
3.4	Build MXD2map	12
3.5	Run MXD2map	12
3.6	Install Mapserver from Source	12
3.7	Setup Mapserver with Apache	14
4	Functionality	15
4.1	How to use	16
4.2	Features	16
5	Restrictions	19
6	Indices and tables	21
	Index	22

LICENCE

GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

4. Do one of the following:

- 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit,

the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by

the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

INSTALLATION FROM SOURCE ON WINDOWS

This Documentation describes how you can install MXD2Map from source. If you only want to use MXD2Map it is strongly recommended that you use our convenient binary installer.

2.1 Requirements

Currently MXD2Map only runs on Windows, because the ArcObjects JDK API does not work properly on GNU/Linux systems. All Free Software components of MXD2Map can run on GNU/Linux as well as on Windows.

To run the converter you need at least the following components:

- Sun Java6 SDK to build, JRE to run
- ArcGIS Desktop10.x or ArcEngine10.x with a valid licence enabled
- UMN MapServer with Java MapScript-bindings ($\geq 6.0.1$ or recent development-version)

Additionally the following components are recommended:

- Apache2 Webserver for displaying the resulting Mapfiles
- MapServer compiled with ArcSDE-plugin
- GDAL/OGR compiled with ArcSDE-support
- GDAL/OGR compiled with FileGeodatabase-support
- Other GDAL/OGR-related tools
- Sphinx for building the docs

2.2 Install Apache Webserver

To display the resulting mapfiles directly you need to install a webserver like Apache2 and the UMN MapServer.

Important: If you are using MS4W, the following installation instruction does apply. , although it can be used to preview the *resulting* mapfiles MS4W does not provide the needed version to run MXD2map.

Download Apache from <https://httpd.apache.org/download.cgi> and install it by executing the .msi-file and following the install-dialogs.

Make sure that you install apache as a service so it is available directly after system startup.

2.3 Install UMN MapServer

It is recommended to use the latest binary release version from <http://www.gisinternals.com/sdk/> as the MapServer for MXD2Map.

1. Unzip the MapServer ZIP-file at `c:\mapserver`
2. Modify the Apache configuration (e.g. `c:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf`) by adding:

```
ScriptAlias /cgi-bin/ "C:/mapserver/bin/"

<Location "/cgi-bin">
    Options None
    Order allow,deny
    Allow from all
</Location>

SetEnv PROJ_LIB "c:/mapserver/bin/proj/SHARE"
SetEnv PATH "c:/mapserver/bin"
SetEnv PATH "c:/mapserver/bin/gdal/python/osgeo"

# set GDAL_DATA environment variable to location of supporting gdal files
SetEnv GDAL_DATA "c:/mapserver/bin/gdaldata"

# set GDAL_DRIVER_PATH environment variable for gdal plugins
SetEnv GDAL_DRIVER_PATH "c:/mapserver/bin/gdal/plugins"
```

3. Copy `mapserv.exe` from `c:\mapserver\bin\ms\apps` to `c:\mapserver\bin\`
4. Optional: Add additional Libraries (DLL's) to `c:\mapserver\bin` for Oracle, SDE etc.

2.4 Install the Converter

The following steps assume the runtime folder to be `c:mxd2map`.

1. Download the Source Package from <https://mxd2map.org/download.html>

2. Extract the tarball using 7-Zip (<http://7-zip.org>) or something similar to `c:\mxd2map\`.
3. Create folder `c:\mxd2map\lib\`
4. Create folder `c:\mxd2map\logs\`
5. Get log4j from <http://logging.apache.org/log4j/1.2/download.html>
6. Get jargs <http://jargs.sourceforge.net/> (Version 1.0)
7. Get apache commons codec from https://commons.apache.org/codec/download_codec.cgi
8. Copy the jar-files of log4j, jargs and commons codec to `c:\mxd2map\lib\`
9. Remove the version numbers of the jar file names in lib, or adopt the build.xml accordingly!
10. Copy `mapscript.jar` and `mapscript.dll` from your MapServer Installation (e.g. `c:\mapserver\bin\ms\java`) to `c:\mxd2map\lib\`
11. Copy the `converter.properties.sample` file to `c:\mxd2map\converter.properties` and edit it according to your needs.

2.5 Compile the converter-sources

To compile the sources you need to have a properly configured ant-environment available. ArcGIS SDK provides an ant-installation which suffices:

```
ant jar
```

or:

```
ant jar-norevision
```

2.6 Setup environment-variables

For running the converter it is necessary to set up some environment variables and adjust the path accordingly.

Adapt the following example script to set the minimum variables needed:

```
@echo off
rem #####EDIT HERE#####
rem Set the following Path variables according to your installation:
set MXD2MAP_PATH=c:\mxd2map
set MAPSERVER_PATH=c:\mapserver
set ARCGISSDK_PATH=c:\Program Files\ArcGIS\DeveloperKit10.0
set ARCGISENGINE_PATH=c:\Program Files\ArcGIS\Engine10.0
```

```

rem No editing should be necessary below this line (in an ideal world)
rem #####

if "%1" == "setenv" goto setenv

%comspec% /k SDKShell.bat setenv %1
goto exit

:setenv
@echo Setting environment for using the GDAL and MapServer tools.

if "%2"=="hideoci" goto hideoci

set ocipath=0
set _path="%PATH:;=" "%"
for %%p in (%_path%) do if not "%~p"==" " if exist %%~p\oci.dll set ocipath=1

if "%ocipath%"=="0" goto hideoci
@echo WARNING: If you encounter problems with missing oci libraries then type:
@echo   SDKShell hideoci
goto setenv2

:hideoci
@echo Hiding the OCI plugin library.
if not exist %GDAL_PATH%\bin\gdal\plugins-optional ^
    mkdir %GDAL_PATH%\bin\gdal\plugins-optional
if exist %GDAL_PATH%\bin\gdal\plugins\ogr_OCI.dll ^
    move %GDAL_PATH%\bin\gdal\plugins\ogr_OCI.dll ^
    %GDAL_PATH%\bin\gdal\plugins-optional\ogr_OCI.dll
if exist %GDAL_PATH%\bin\gdal\plugins\gdal_GEOR.dll ^
    move %GDAL_PATH%\bin\gdal\plugins\gdal_GEOR.dll ^
    %GDAL_PATH%\bin\gdal\plugins-optional\gdal_GEOR.dll

:setenv2
SET PATH=%ARCGISSDK_PATH%\java\jre\bin;%ARCGISENGINE_PATH%\bin;^
%MXD2MAP_PATH%\bin;%MAPSERVER_PATH%\bin\gdal\python\osgeo;^
%MAPSERVER_PATH%\bin\proj\apps;%MAPSERVER_PATH%\bin\gdal\apps;^
%MAPSERVER_PATH%\bin\ms\apps;%MAPSERVER_PATH%\bin\gdal\csharp;^
%MAPSERVER_PATH%\bin\ms\csharp;%MAPSERVER_PATH%\bin\curl;^
%MAPSERVER_PATH%\bin\ms\java;%PATH%
SET GDAL_DATA=%MAPSERVER_PATH%\bin\gdal-data
SET GDAL_DRIVER_PATH=%MAPSERVER_PATH%\bin\gdal\plugins
SET PYTHONPATH=%MAPSERVER_PATH%\bin\gdal\python\osgeo;%PYTHONPATH%
SET PROJ_LIB=%MAPSERVER_PATH%\bin\proj\SHARE

:exit

```

Save the above script as `c:\mxd2map\SDKShell.bat` and execute it before working with MXD2map.

2.7 Configure Logging

Logging is supported on different levels. MXD2map logs with Log4J. A sample configuration for Log4J could look like this:

```
#### Log just errors and warnings to a file.
log4j.rootLogger=DEBUG, MXD, Console

log4j.appender.MXD=org.apache.log4j.RollingFileAppender
log4j.appender.MXD.File=./logs/console-client.log
log4j.appender.MXD.MaxFileSize=5000KB
log4j.appender.MXD.MaxBackupIndex=1
log4j.appender.MXD.layout=org.apache.log4j.PatternLayout
log4j.appender.MXD.layout.ConversionPattern=%d [%t] %-5p %c - %m%n

#log4j.rootLogger=WARN, Console
log4j.appender.Console=org.apache.log4j.ConsoleAppender
log4j.appender.Console.Threshold=INFO
log4j.appender.Console.layout=org.apache.log4j.PatternLayout
log4j.appender.Console.layout.ConversionPattern=%d [%t] %-5p %c - %m%n
```

2.7.1 Apache Webserver and UMN MapServer

They both support logging. MapServers output ends up in Apaches logfiles if not configured differently within a providing template. The loglevel can be adjusted at template-level. For further information consult the UMN MapServer manual regarding logging at <http://mapserver.org/optimization/debugging.html#steps-to-enable-mapserver-debugging>.

INSTALLATION FROM SOURCE ON GNU/LINUX

This Documentation describes how you can install MXD2Map from source on GNU/Linux Systems.

Download the Source tarball of mxd2map and extract it into a directory. The Path used in this example is ~/mxd2map.

3.1 Requirements

To run the converter you need at least the following components:

- Sun Java6 SDK / JRE and ant
- ArcEngine10.x with a valid licence enabled
- Some additional Java Tools and libraries as documented below

3.2 Install ArcGIS SDK

This part is a bit out of scope of MXD2map, but here is an example on how the environment for MXD2map could be set up for CentOS 5.

Important: Deactivate SELINUX! ArcGIS fails in many mysterious ways otherwise. to do this write SELINUX=permissive into: /etc/sysconfig/selinux and run “setenforce Permissive”

1. Install the dependencies:

```
yum-install compat-libstdc++-33 compat-libstdc++-296 compat-libf2c compat-lib  
compat-openldap-2.3 cairo compat-libf2c-34 mesa-libGLU compat-gcc-  
libidn mesa-libGL mesa-libGLU openldap openmotif-devel openmotif  
yum-groupinstall "Legacy Software Support"  
yum-groupinstall "X Window System"  
yum-groupinstall "Legacy Software Development"
```

```
yum-groupinstall "X Software Development"
yum-groupinstall "Development Tools"
```

This actually installs way to much but sadly only those large group requirements were documented by ESRI.

1. Create a User and a paths for arcgis:

```
useradd mxd2map
groupadd arcgis-users
usermod -a -G arcgis-users mxd2map
mkdir -p /usr/local/share/macrovision/storage
chmod g+w /usr/local/share/macrovision/storage
chgrp arcgis-users /usr/local/share/macrovision/storage
```

2. Set passwd for user mxd2map and log in as that user.
3. Mount the ArcGIS-SDK DVD under /mnt/ArcGIS-SDK and run

```
/mnt/ArcGIS-SDK/linux/EngineRT/Setup
```

4. Click through the dialog leave everything as default. Then run:

```
/mnt/ArcGIS-SDK/linux/ArcObjectsSDKJava/Setup
```

5. Click through the dialog leave everything as default
6. Register “ArcGIS Engine Runtime and ArcGIS Engine Developer Kit (Single Use)”

3.3 Install additional libraries for mxd2map

Assuming that you have mxd2map extracted in ~/mxd2map you can execute the following to install jargs, log4j and apache commons-codec:

```
cd ~/mxd2map
mkdir lib
# Jargs
wget http://sourceforge.net/projects/jargs/files/jargs/1.0/jargs-1.0.tar.bz2/download
tar -xj jargs.jar -O > jargs.jar

# log4j
wget http://archive.apache.org/dist/logging/log4j/1.2.9/logging-log4j-1.2.9.tar.gz
tar -xz log4j-1.2.9.jar -O > log4j.jar

# Commons codec
wget http://apache.openmirror.de/commons/codec/binaries/commons-codec-1.7-bin.tar.gz
tar -xz commons-codec-1.7.jar -O > commons-codec.jar
```

3.4 Build MXD2map

You need to setup some environment variables for ArcGIS first:

```
export JAVA_HOME=/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0
. ~/arcgis/engine10.0/init_engine.sh
. ~/arcgis/engine10.0/java/tools/init_java.sh
. ~/arcgis/developerkit10.0/java/tools/setenv_ant.sh
. ~/arcgis/developerkit10.0/init_devkit.sh
cd ~/mxd2map
ant jar-norevision
```

3.5 Run MXD2map

ArcGIS needs some weird environment settings. Additionally it provides it's own versions of basic libraries like libstdc++. Here is what you have to do before you can start mxd2map:

```
export JAVA_HOME=/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0
. ~/arcgis/engine10.0/init_engine.sh
. ~/arcgis/engine10.0/java/tools/init_java.sh
. ~/arcgis/developerkit10.0/java/tools/setenv_ant.sh
. ~/arcgis/developerkit10.0/init_devkit.sh
export LD_LIBRARY_PATH=$DEVELDIR/lib:$LD_LIBRARY_PATH
export LD_PRELOAD=$LD_PRELOAD:/usr/lib/libstdc++.so.6
export AWT_TOOLKIT=XToolkit
export LC_ALL=en_US.Latin1
```

Now edit the converter.properties sample so that it points to a valid mxd file together with a valid mapfile template.

Note: If your files contain special characters in windows encoding convert them with convmv:

```
convmv --notest -r -fcp1252 -t utf8 Bereisungen/
```

Important: The default font file packaged with mxd2map contains the windows paths to the fonts. To set them to the linux paths you can run:

```
sed -i 's/C:\Windows\Fonts\/\home\/mxd2map\/arcgis\/engine10.0\/fonts/' ttf-fo
```

3.6 Install Mapserver from Source

This documentation is a log of the steps necessary to build mapserver 6.2.0-rc1 on CentOS 5. You might just want to install a binary package for your distribution For a real documentation please refer to: <http://mapserver.org/installation/unix.html>

Here is how you install a mapserver into ~/devel:

```
# As root:
yum groupinstall "Development Tools"
yum install freetype-devel libpng-devel zlib-devel curl-devel libtiff-devel libj
libXpm-devel libxml2-devel expat-devel sqlite-devel agg-devel giflib-devel libx

# As user:
# set up some convenience functions
export DEVELDIR=~/.devel
export SRCDIR=$DEVELDIR/src
alias configure="./configure --prefix=$DEVELDIR"
function cs {
    mkdir -p "$SRCDIR"
    if test -n "$1"; then
        cd "$SRCDIR/$1"
    else
        cd "$SRCDIR"
    fi
}

cs
wget http://download.osgeo.org/proj/proj-4.8.0.tar.gz
tar -xf proj-4.8.0.tar.gz
cd proj-4.8.0
configure && make && make install

cs
wget https://bitbucket.org/pierrejoye/gd-libgd/get/GD_2_0_33.tar.bz2
tar -xf GD_2_0_33.tar.bz2
cd pierrejoye-gd-libgd-5551f61978e3/src
configure && make && make install

cs
wget http://download.osgeo.org/geotiff/libgeotiff/libgeotiff-1.4.0.tar.gz
tar -xf libgeotiff-1.4.0.tar.gz
cd libgeotiff-1.4.0
configure --with-proj=$DEVELDIR && make && make install

cs
wget http://downloads.esri.com/Support/downloads/ao_/FileGDB_API_1_2-32.tar.gz
tar -xf FileGDB_API_1_2-32.tar.gz
cd FileGDB_API_1_2-32
export JAVA_HOME=/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0
. ~/arcgis/engine10.0/init_engine.sh
. ~/arcgis/engine10.0/java/tools/init_java.sh
. ~/arcgis/developerkit10.0/java/tools/setenv_ant.sh
. ~/arcgis/developerkit10.0/init_devkit.sh

cs
wget http://download.osgeo.org/gdal/gdal-1.9.2.tar.gz
```



```
tar -xf gdal-1.9.2.tar.gz
cs gdal-1.9.2
configure --enable-shared=no --enable-static=yes --with-fgdb=$DEVELDIR && make &

cs
wget http://download.osgeo.org/mapserver/mapserver-6.2.0-rc1.tar.gz
tar -xf mapserver-6.2.0-rc1.tar.gz
cs mapserver-6.2.0-rc1
configure -with-proj=$DEVELDIR --with-gd=$DEVELDIR \
  --with-ogr=$DEVELDIR/bin/gdal-config \
  --with-gdal=$DEVELDIR/bin/gdal-config \
  --with-wfs --with-wcs --with-wmsclient --with-wfsclient --with-sos \
  --with-xml2-config=/usr/bin/xml2-config --with-apache-module \
  --with-apxs=/usr/sbin/apxs --with-exslt=yes --with-xslt=yes
# in this specific version you also need to apply https://github.com/unicolet/ma
# when building on x86 systems
make && make install
cd mapscript/java
make && make install
cp mapscript.jar ~/mxd2map/lib
```

3.7 Setup Mapserver with Apache

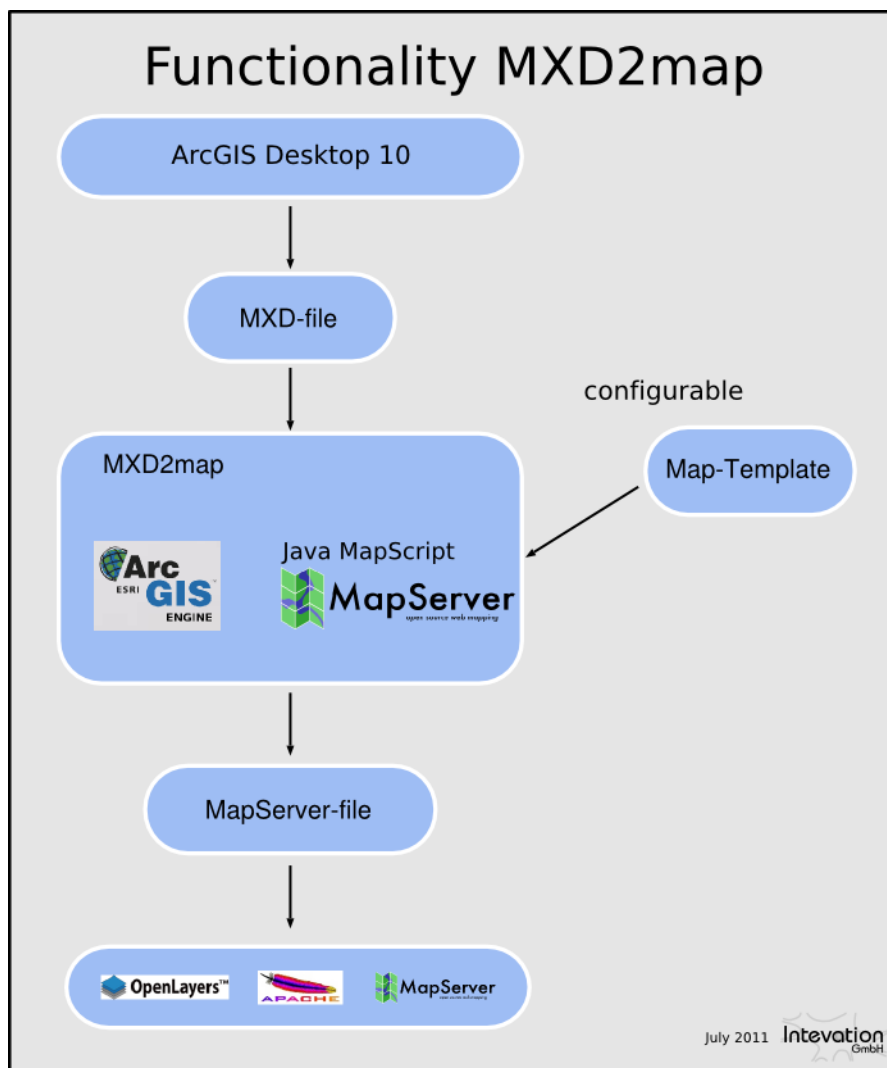
Edit `/etc/httpd/conf/httpd.conf` and change the setting of the directory `/var/www/cgi-bin`:

```
<Directory "/var/www/cgi-bin">
  AllowOverride None
  Options ExecCGI
  Order allow,deny
  Allow from all
</Directory>
```

FUNCTIONALITY

The MXD2map converter works as a commandline tool to transform maps from the proprietary ArcGIS file format “MXD” to a MapServer configuration file.

To transform an ArcGIS mxd file to a MapServer configuration file the mxd file must have a valid and available data source and must not fit the *Restrictions* described in the appropriate section.



4.1 How to use

The converter is started via the commandline running the command:

```
$java -jar mxd2map.jar
```

The converter can be configured in two ways:

1. The `converter.properties` file (see `converter.properties.sample`)
2. via commandline arguments

The `converter.properties.sample` file:

```
# Java properties file for the MXD Converter Tool.
mxd = \\full\\path\\to\\mxd-file
map = \\full\\path\\to\\map-file
map-template = \\full\\path\\to\\mapfile-template
```

Make sure you mask backslashes with an additional backslash.

Both take a parameter for the path to the mxd file, the outputfile and a mapfile template for initial mapfile config. Commandline parameters have a higher priority than the properties file.

The available parameters on the commandline are:

```
-m or --mxd           The mxd file to convert.
-a or --map           The path to the output file. Should end with ".map".
-t or --template      The template to use for an initial mapfile
                      configuration.
```

To have the full functionality for ArcGIS map conversion it is necessary to have the ESRI fonts available. ArcGIS products install these fonts to the default windows font folder `c:\\windows\\fonts`. To make them available for the converter there has to be a MapServer font set (see <http://mapserver.org/mapfile/fontset.html>).

The MXD2map converter comes with an initial fontset containing all relevant ESRI fonts and the free font FreeSans. The font set file named `fonts.txt` can be edited and further fonts can be added. To use a font set, the file needs to be referenced in the MapServer template otherwise the converter can not create character symbols and will show up errors in the conversion process.

The converter uses a separate symbol file to provide symbol sets to the template to have initial symbols available. If no symbol file is referenced a new one is created by the converter if necessary. The resulting symbol set is written to the same directory as the output mapfile named `[mapfile-name]-symbols.sym`.

4.2 Features

MXD2map is optimized to work best with MapServer 6.x. Therefore the new MapServer features (symbolhandling, styling etc) are built in.

MXD2map currently supports the following features:

- Datasources:
 - ArcSDE vector database connections (requires MapServer built against ArcSDE, Client-dlls [`sde.dll`, `sg.dll` and `pe.dll`] needed)
 - ESRI-Shapefiles
- Symbols:
 - Simple marker symbols
 - Character marker symbols
 - Arrow marker symbols
 - Picture marker symbols (also transparency is supported if set in MXD)
 - Symbols based on TTF-characters of special cartographic fonts
 - Any combination of character marker symbols
 - Linesymbols with pattern and cartographic attributes
 - Polygons filled with solid color and outline
 - Polygons filled with a hatch and hatched outline
- Layer types:
 - Featurelayer (vectors) as POINT, LINESTRING and POLYGON
 - Classifications with unique values
 - Classifications with class breaks
 - Group-members are concatenated to `group.layername`. This ensures having discrete layernames
 - ArcGIS-layergroups are represented as WMS Named layers, which are WMS-layers without a title (so not requestable through GetMap)
 - all definition-queries for filtering the data on Layer- and classification-level
- Further map attributes:
 - Projections and units based on the EPSG-table (if set in MXD)
 - Min/max scale denominations at layer and label-level
 - Global Map extent
 - Layer extent and SRS
 - Simple labeling
 - Filter and expressions
 - ArcSDE Jointables
 - Mapfile-Templating for OGC-related stuff

- Simple Labeling-syntax (a subset of ArcGISs VBScript-labels)
- Other features
 - Umlauts are translated into its equivalents within LAYER-names, CLASS-names and also mapfile-names. Though it is good standard not to use them since they mostly cause problems when using them within OWS-related services.
 - Creation of distinct layer-names by concatenating group- and layernames to WMS-titles. This comes handy mostly in ArcGIS layergroups.

RESTRICTIONS

Currently there are some restrictions in the usage of the converter. One of the main restrictions is the fact that the converter currently only runs on Windows. This is due to the fact that the needed ArcEngine did never reliably work on Linux. This was tested with ArcEngine 10.0.

Supported datasources:

- ArcSDE-passwords cannot be read since they are encrypted. The converter exports the string <PASSWORD> which needs to be replaced later on in the resulting mapfile.
- Since MapServer cannot use a FileGeodatabase datasources (yet) the MXD2map does not support this datasource and will not transform the MXD-file at all. This may change in the near future.
- Raster data is not supported by the converter yet, though ArcSDE Raster-support will be available in the near future.

Symbol support:

- Line symbols with pattern cannot be combined with marker symbols. The converter reads all symbols and writes the corresponding styles to the resulting mapfile but the marker symbols are rendered without any gap. The resulting mapimage may not look as intended.

Hint: Workarround: For small marker symbols it is recommended to use a line symbol in combination with pattern (in ArcGIS called template) to create the symbology. The reason why marker symbols cannot be used in combination with a template is the difficulty to interpret the read attributes like marks and gaps and write the appropriate symbols with the appropriate gap to the mapfile.

- The converter sets the width to 1 for line symbols that have a width smaller than 1.
-

Tip: To avoid lines and line colors (specially outlines) to be handled by the converter, use integer numbers for line width only.

- In ArcGIS maps default symbols can be defined as fallback within classifications. Since MapServer does not support such fallback symbols or styles within classifications the converter does not read these default symbols.

- In ArcGIS predefined symbols like bars or pie charts can be used to display values. The converter does not support these symbols yet.

Layer support:

- Expressions for label are parsed by an internal parser in ArcGIS. The interpretation of this expressions is very limited though. Therefore label expressions may not contain any special functions or must fit the mapserver expression design pattern.

INDICES AND TABLES

- *genindex*
- *search*

INDEX

A

apache, 4, 9
ArcEngine, 18
ArcGIS, 9
ArcSDE, 4, 18

B

building, 9

C

CentOS, 9
commandline, 14
compile, 4, 9
configuration, 14
contents, 1

F

features, 14
functionality, 14

I

installation, 4, 9

J

jar, 4

L

label, 18
LGPL, 1
licence, 1
linux, 9
logging, 4

P

picture-symbols, 18

R

Raster, 18

requirements, 4, 9

restrictions, 18

runtime-parameter, 14

S

SDE, 18

U

UMN MapServer, 4, 9

W

windows, 18

workaround, 18