

MapInfo MRR – Multi Resolution Raster GDAL Driver

This driver supports reading of MRR (Multiple Resolution Raster) file format developed by Precisely(MapInfo). This driver does not provide support for creating, writing or editing MRR files.

Driver capabilities

Contents

1. Overview of MRR (Multiple Resolution Raster) Driver
2. Issues and Limitations
3. Download MRR Plugin and MapInfo Pro Advanced SDK Runtime
4. Building MRR driver from source
5. Configure QGIS to use mrr plugin

Overview of MRR (Multiple Resolution Raster) Driver

MRR unifies the storage of all types of raster data such as imagery, spectral imagery, continuous gridded data and thematic data. MRR extends the concept of a multi-banded raster to a “four dimensional” raster which may contain –

- One or more fields, each of which contain a particular type of raster data. A field may contain multi-banded continuous grid data, multi-banded classified data, color imagery or color imagery utilizing a fixed size color palette.
- One or more events, each of which contain an addition or modification to the field data at a specified time. Events provide a time dimension in MRR.
- One or more bands which contain data in one of many supported data types such as 1/2/4/8/16/32/64 bit integers and 32/64 bit floating point. Some data types, like color or complex numbers, may contain multiple components. These are exposed as virtual bands.
- A stack of overviews, referred to as resolutions levels. Level 0 contains the ‘base resolution’ raster data. Levels 1 upward contain overviews within which the cell size doubles at each level. Levels -1 downward contain underviews which are generated on demand by interpolation from the base level.

- MRR is designed to enable the creation of very large and high resolution rasters and the SDK ensures that access to data at any resolution level is constant. Raster data is stored in a sparse collection of tiles of equal size. Lossless compression codecs are employed to store data within each tile, and lossy image compression codec can be used for imagery fields. Each resolution level has a fixed cell size, but MRR supports a multi-resolution tile concept which allows the cell size to set in each tile separately.

NOTE: Some MRR features may not be accessible through GDAL driver.

The MRR driver for GDAL is supported on 64-bit Windows, Ubuntu, Amazon-Linux, OracleLinux and CentOS.

Issues and Limitations

- The driver does not provide support for creating, writing or editing MRR files.
- Although an MRR may contain multiple fields, this driver can only access the first field.
- Although an MRR may contain multiple events, this driver can only access data that represents the roll-up of all events. This represents the “final state” of the raster.

Download MRR Plugin and MapInfo Pro Advanced SDK Runtime

MRR driver needs MapInfo Pro Advanced SDK to be installed on the machine to work with GDAL.

Download MapInfo Pro Advanced SDK Runtime and GDAL plugin by navigating to this link “<https://www.pbinsight.com/support/product-downloads/item/mapinfo-mrr-sdk-download>”. Once the package is downloaded, unzip the package on your machine at the desired location.

|

|

GDAL_MRR Zip folder Structure:

GDAL_MRR Zip file contains following folders:

- bin -> 64bit binaries for gdal mrr plugin and mrr SDK Runtime
 - AmazonLinux -> 64 bit Binaries for AmazonLinux
 - * MRR_Plugin -> gdal_MRR.so plugin binary
 - * MRR_SDK_Runtime -> binaries for MRR SDK runtime
 - CentOS7 -> 64 bit Binaries for CentOS7
 - * MRR_Plugin -> gdal_MRR.so plugin binary
 - * MRR_SDK_Runtime -> binaries for MRR SDK runtime
 - OracleLinux -> 64 bit Binaries for OracleLinux
 - * MRR_Plugin -> gdal_MRR.so plugin binary
 - * MRR_SDK_Runtime -> binaries for MRR SDK runtime
 - Ubuntu -> 64 bit Binaries for Ubuntu
 - * MRR_Plugin -> gdal_MRR.so plugin binary
 - * MRR_SDK_Runtime -> binaries for MRR SDK runtime
 - Windows -> 64 bit Binaries for Windows
 - * MRR_Plugin -> gdal_MRR.dll plugin binary
 - * MRR_SDK_Runtime -> binaries for MRR SDK runtime
- mrr -> Source code to build MRR plugin when needed

Follow these two simple steps to configure GDAL MRR plugin on your system:

1. Copy gdal_MRR.so or gdal_MRR.dll as per your platform to GDAL_DRIVER_PATH folder (you will need to explicitly set this environment variable to allow GDAL to find its plugin drivers)
 - a. **Linux Example:** -export GDAL_DRIVER_PATH=/usr/local/mrrdrv
-cp gdal_MRR.so \$GDAL_DRIVER_PATH
 - b. **Windows Example:** -SET GDAL_DRIVER_PATH=C:mrrdrv
-cp gdal_MRR.dll %GDAL_DRIVER_PATH%
 - c. Alternatively, you could place it under /usr/local/lib/gdalplugins directory, and (if known) the lib/gdalplugins subdirectory of the gdal home directory on Linux and \$(BINDIR)gdalplugins on Windows. (This does not always work options a, b is recommended)

2. Copy MRR SDK runtime bins from appropriate folder as per your platform into the folder which contains GDAL bins.
 - a. Example `cp /GDAL_MRR/bin/Ubuntu/ /usr/local/lib/`
 - b. `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib/`
-

Building MRR driver from source

MRR driver for popular versions of GDAL is available in plugin form for most platforms. In case you don't find your Platform/GDAL version of choice from this pre-build set of binaries, you may follow the instructions below to build it on your own, it's really easy. You may also want to refer to official GDAL sites for this purpose:

<https://gdal.org/>

<https://trac.osgeo.org/gdal/wiki/BuildHints>

<https://github.com/OSGeo/gdal>

Building plugin gdal mrr driver is two step process:

First you get appropriate version of GDAL source from gdal git repo mentioned above. Then build gdal driver as you would usually using below steps.

Build GDAL driver: Linux:

1. Switch to GDAL root folder e.g. `cd /usr/local/data/gdal/`
2. `make clean`
3. `configure build : ./configure --without-libtool`
4. `make`
5. `make install`

Note: *GCC version 4.9 is preferred, as MRR SDK runtime is built with GCC 4.9*

Windows:

You will need Proj 6.0 or greater libraries before you proceed with building GDAL. You could download it from below location:

http://download.osgeo.org/osgeo4w/osgeo4w-setup-x86_64.exe

Now open `nmake.opt` file from gdal root folder (e.g. `c:\gdal`) in text editor and Change below variables:

1. **Update GDAL_HOME location** e.g. GDAL_HOME = "C:\gdal"
2. **Update PROJ_INCLUDE and PROJ_LIBRARY as per your machine installation.**
e.g. enable below two lines


```

PROJ_INCLUDE    =    -IC:\OSGeo4W64\apps\proj-
dev\include
PROJ_LIBRARY    =    C:\OSGeo4W64\apps\proj-
dev\lib\proj_6_1.lib

```

Now open "x64 Native Tools command prompt of Visual Studio" and issue following commands to build GDAL

1. navigate to gdal root folder e.g. cd c:\gdal
2. "nmake /f makefile.vc WIN64=1" to build GDAL and apps for win64 with msvc VS2015.
3. "nmake /f makefile.vc WIN64=1 install" to install GDAL

Build MRR Plugin driver: To build MRR driver as plugin(autoload) driver.

Copy mrr folder from GDAL_MRR.zip as it is to your gdal location e.g. ../gdal/frmts/

switch to /gdal/frmts/mrr directory and issue following build commands based up on your platform.

Linux:

1. switch to mrr directory e.g. cd /usr/local/data/gdal/frmts/mrr
2. make plugin

Windows:

1. open "x64 Native Tools command prompt of Visual Studio" and switch to mrr directory e.g. c:\gdal\frmts\mrr
2. nmake /f makefile.vc plugin

Validate configuration: Following commands could be used to check whether MRR plugin configuration:

- gdalinfo -format mrr
- gdalinfo /path/to/some/mrrfile.mrr

Configure QGIS to use mrr plugin

1. Navigate to gdalplugins folder of QGIS (Windows e.g. C:\Program Files\QGIS 3.10\bin\gdalplugins) and drop gdal_MRR.dll there
2. drop mrr SDK runtime bins to QGIS bin folder (e.g. C:\Program Files\QGIS 3.10\bin)

QGIS should now be able to discover and work with MapInfo MRR format.