

# Tutorial: compile TerraLib usando MinGW



Leandro da Silva Santos  
leandrosilvasantos@gmail.com

Tiago Garcia de Senna Carneiro  
tiago@iceb.ufop.br

Departamento de Computação  
Universidade Federal de Ouro Preto



UFOP

Universidade Federal  
de Ouro Preto

Laboratório Associado INPE/UFOP para Modelagem e Simulação de Sistemas Terrestres

---

## Pré-requisitos:

- Sistema Operacional Windows.
- Compilador MinGW.
- Eclipse com CDT.
- Biblioteca QT instalada.

## Procedimento:

1 - Faça o *download* do seguinte arquivo [TerraLib\\_linux\\_v\\_3\\_1\\_4.tar.gz](#) no site do [Terralib](#).

2 - Extraia o conteúdo do arquivo [TerraLib\\_linux\\_v\\_3\\_1\\_4.tar.gz](#).

2.1- Contém duas pastas [src](#) e [terralibx](#).

3 - Crie um projeto *Standard Make C++ Project* no Eclipse.

3.1 - Selecione o Projeto com o mouse.

3.2 - Selecione no *menu* a opção *Project*, em seguida *Properties*.

3.3 - Selecione na janela *Properties* for <nome do projeto> a opção *C/C++ Make Project*.

3.4 - Selecione a aba *Make Builder*, desmarque a opção *Use default* em seguida coloque no campo *Build command* o comando `mingw32-make -f makefile`.

3.5 – Clique em *OK*.

4 - Importe para o projeto as pasta [src](#) e [terralibx](#).

5 - Importe para o projeto a pasta [terralibx](#) - que está disponível para download no link: [terralibw.zip](#). Sobrescreva os arquivos.

6 - Selecione o projeto com o mouse

6.1 - Selecione no menu a opção *Project*, em seguida *Properties*.

6.2 - Selecione na janela *Properties for <nome do projeto>* a opção *C/C++ Make Project*.

6.3 - Selecione a aba *Make Builder*, no campo *Build directory* localize no *workspace* a pasta “terralibx/libjpeg”.

6.4 - Clique em OK.

7 - Aplique o comando `qmake -o makelibJpeg libjpeg.pro` via linha de comando na pasta “terralibx/libjpeg”.

8 - Abra o arquivo makelibJpeg

8.1 - Substitua a linha de comando

```
DESTDIR_TARGET = ..\lib\libjpeg.lib
```

por

```
DESTDIR_TARGET = ..\lib\jpeg.a
```

8.2 - Substitua a linha de comando

```
$(LINK) $(LFLAGS) -o "$(DESTDIR_TARGET)" object_script.jpeg $(LIBS)
```

por

```
ar rc "$(DESTDIR_TARGET)" obj\jcapimin.o obj\jcapistd.o obj\jccoefct.o obj\jccolor.o obj\jcdctmgr.o \
obj\jchuff.o obj\jcinit.o obj\jcmaint.o obj\jcmarker.o obj\jcmaster.o obj\jcomapi.o obj\jcparm.o obj\jcphuff.o \
obj\jcpresct.o obj\jcsample.o obj\jctrans.o obj\jdapimin.o obj\jdapistd.o obj\jdatadst.o obj\jdatasrc.o obj\jdcoefct.o \
obj\jdcolor.o obj\jddctmgr.o obj\jdhuft.o obj\jdinpnt.o obj\jdmainct.o obj\jdmarker.o obj\jdmaster.o obj\jdmerge.o \
obj\jdpuff.o obj\jdpstct.o obj\jdsample.o obj\jdtrans.o obj\jerror.o obj\jfdctflt.o obj\jfdctfst.o obj\jfdctint.o \
obj\jidctflt.o obj\jidctfst.o obj\jidctint.o obj\jidctred.o obj\jmemmgr.o obj\jmemnobs.o obj\jquant1.o obj\jquant2.o \
obj\jutils.o obj\rdbmp.o obj\rdcolmap.o obj\rdgif.o obj\rdppm.o obj\rdrl.o obj\rdswitch.o obj\rdtarga.o obj\transupp.o \
obj\wrbmp.o obj\wrgif.o obj\wrppm.o obj\wrrle.o obj\wrtarga.o
```

8.3 - Faça o Build.

10 - Na pasta “terralibx/lib” deverá contém a biblioteca libjpeg.a.

11 - Selecione o projeto com o mouse

11.1 - Selecione no menu a opção *Project*, em seguida *Properties*.

11.2 - Selecione na janela *Properties for <nome do projeto>* a opção *C/C++ Make Project*.

11.3 - Selecione a aba *Make Builder*, no campo *Build directory* localize no *workspace* a pasta “terralibx/tiff”.

11.4 - Clique em OK.

12 - Aplique o comando `qmake -o makeTiff tiff.pro` via linha de comando na pasta “terralibx/tiff”.

13 - Abra o arquivo makelibTiff

13.1 - Substitua a linha de comando

```
DESTDIR_TARGET = ..\lib\libtiff.lib
```

por

```
DESTDIR_TARGET = ..\lib\libtiff.a
```

13.2 - Substitua a linha de comando

```
$(LINK) $(LFLAGS) -o "$(DESTDIR_TARGET)" object_script.tiff $(LIBS)
```

por

```
ar rc $(DESTDIR_TARGET) obj\cpl_csv.o obj\cpl_serv.o obj\fax3sm.o obj\geo_extra.o \
obj\geo_free.o obj\geo_get.o obj\geo_names.o obj\geo_new.o obj\geo_normalize.o obj\geo_print.o obj\geo_set.o \
obj\geo_tiffp.o obj\geo_trans.o obj\geo_write.o obj\geotiff_proj4.o obj\tif_aux.o obj\tif_close.o obj\tif_codec.o \
obj\tif_color.o obj\tif_compress.o obj\tif_dir.o obj\tif_dirinfo.o obj\tif_dirread.o obj\tif_dirwrite.o obj\tif_dumpmode.o \
obj\tif_error.o obj\tif_extension.o obj\tif_fax3.o obj\tif_flush.o obj\tif_getimage.o obj\tif_jpeg.o obj\tif_luv.o \
obj\tif_lzw.o obj\tif_machdep.o obj\tif_next.o obj\tif_open.o obj\tif_packbits.o obj\tif_pixarlog.o obj\tif_predict.o \
obj\tif_print.o obj\tif_read.o obj\tif_strip.o obj\tif_swab.o obj\tif_thunder.o obj\tif_tile.o obj\tif_unix.o obj\tif_version.o \
obj\tif_warning.o obj\tif_write.o obj\tif_zip.o obj\xtiff.o
```

### 13.3 - Faça o *Build*.

14 – Na pasta “terralibx/lib” deverá conter a biblioteca libtiff.a.

15 - Selecione o projeto com o mouse

15.1 - Selecione no menu a opção *Project*, em seguida *Properties*.

15.2 - Selecione na janela *Properties for <nome do projeto>* a opção *C/C++ Make Project*.

15.3 - Selecione a aba *Make Builder*, no campo *Build directory* localize no *workspace* a pasta “terralibx/terralib”.

15.4 - Clique em *OK*.

16 - Aplique o comando `qmake -o makeTerralib terralib.pro` via linha de comando na pasta “terralibx/terralib”.

17 - Abra o arquivo makeTerralib.

17.1 - Substitua a linha de comando

```
DESTDIR_TARGET = ..\lib\libterralib.lib
```

por

```
DESTDIR_TARGET = ..\lib\libterralib.a
```

17.2 - Substitua a linha de comando

```
$(LINK) $(LFLAGS) -o "$(DESTDIR_TARGET)" object_script.libterralib $(LIBS)
```

por

```
ar rc $(DESTDIR_TARGET) obj\lexTemporal.o obj\TeAffineGT.o obj\TeAffineGTFactory.o \
obj\TeAsciiFile.o obj\TeBox.o obj\TeBufferRegion.o obj\TeCentroid.o obj\TeComputeAttributeStrategies.o \
obj\TeDatabase.o obj\TeDatum.o obj\TeDecoderASCIIGrid.o obj\TeDecoderDatabase.o obj\TeDecoderFile.o \
obj\TeDecoderJPEG.o obj\TeDecoderMemory.o obj\TeDecoderMemoryMap.o obj\TeDecoderSPR.o \
obj\TeDecoderTIFF.o obj\TeDecoderVirtualMemory.o obj\TeErrorLog.o obj\TeErrorMessage.o obj\TeException.o \
obj\TeFragmentation.o obj\TeGeometricTransformation.o obj\TeGeometry.o obj\TeGeometryAlgorithms.o \
obj\TeGTPParams.o obj\TeGTFactory.o obj\TeGroupingAlgorithms.o obj\TeImportRaster.o obj\TeInitQuerierStrategy.o \
obj\TeInitRasterDecoders.o obj\TeIntersector.o obj\TeLayer.o obj\TeLegendEntry.o obj\TeMatrix.o \
obj\TeMultiGeometry.o obj\TeNeighbours.o obj\TeNetwork.o obj\TeOverlay.o obj\TeOverlayUtils.o obj\TeProgress.o \
obj\TeProjection.o obj\TeProxMatrixConstructionStrategy.o obj\TeProxMatrixImplementation.o \
obj\TeProxMatrixStrategies.o obj\TeQuerier.o obj\TeQuerierDB.o obj\TeQuerierDBStr1.o obj\TeQuerierDBStr2.o \
obj\TeQuerierDBStr3.o obj\TeQuerierImpl.o obj\TeQuerierParams.o obj\TeRaster.o obj\TeRasterParams.o \
obj\TeRasterRemap.o obj\TeRasterTransform.o obj\TeRepresentation.o obj\TeSpatialOperations.o obj\TeStdFile.o \
obj\TeSTElement.o obj\TeSTElementSet.o obj\TeSTEvent.o obj\TeSTInstance.o obj\TeTable.o \
obj\TeTemporalSeries.o obj\TeTheme.o obj\TeTime.o obj\TeTimeInterval.o obj\TeTin.o obj\TeUtils.o \
obj\TeVectorRemap.o obj\TeViewNode.o obj\yyTemporal.o obj\TeDatabaseFactoryParams.o obj\TeLibJpegWrapper.o \
obj\jmemdstd.o obj\jmemsrc.o
```

18 - Faça o *download* do arquivo no link [mysql-mingw.zip](#).

19 - Importe os arquivos libmySQL.a e libmySQL.def para a pasta “terralibx/lib”.

20 - Faça o *Build*.

21 - Na pasta “terralibx/lib” deverá contém a biblioteca libterralib.a.

---

## Compilação do exemplo CreateDatabase

### Pré-requisitos:

- Projeto Terralib compilado.
- MySQL instalado.

### Procedimento:

1 - No projeto anterior crie a pasta examples.

2 - Importe para a pasta examples a pasta createDatabase presente no link [createdatabase.zip](#).

4 - Selecione o projeto com o mouse

4.1 - Selecione no menu a opção *Project*, em seguida *Properties*.

4.2 - Selecione na janela *Properties for <nome do projeto>* a opção *C/C++ Make Project*.

4.3 - Selecione a aba *Make Buidler*, no campo *Build directory* localize no *workspace* a pasta “examples/createDatabase”.

4.4 - Clique em *OK*.

5 – Aplique o comando `gmake -o makeCreateDatabase createDatabase.pro` via linha de comando na pasta “examples/createDatabase”.

6 - Faça o *Build*.

9 - Abra o arquivo makeCreateDatabase da pasta “examples/createDatabase”.

9.1 - Substitua a linha de comando first: debug por first: release

10 - Faça o Build.

Obs: Para execução do exemplo é necessário o arquivo libmysql.dll.