Date Modified: March. 28, 2025

**LS-Reader**

1. **Supported Platforms**

\* Microsoft Windows (vs2017, vs2019, vs2022)

\* Linux (GCC >= 4.8)

1. **Supported Languages**

\* C++

\* C

\* Python (3.6, 3.7, 3.8, 3.9, 3.10, 3.11)

1. **Tree structure in the distribution directory**

.

|-- CMakeLists.txt (standard input file for CMake)

|-- Macros.cmake (CMake macros)

|-- README.docx (this file)

|-- config.h.in (data\_path definition for C++ and C)

|-- documents

| |-- LSReader\_C.docx (introductions about APIs and some examples for C)

| |-- LSReader\_CXX.docx (introductions about APIs and some examples for C++)

| `-- LSReader\_Python.docx (introductions about APIs and some examples for Python)

|-- example\_data

| |-- binout

| |-- d3plot

| |-- d3plot01

| |-- d3plot02

|-- lsreadercdist

| |-- CMakeLists.txt

| |-- binoutreaderc.h

| |-- d3plotreaderc.h

| |-- example.c

| `-- lib

| |-- linux

| | |-- liblsreader\_c.so -> liblsreader\_c.so.0.1.0

| | `-- liblsreader\_c.so.0.1.0

| |-- vs2010

| | |-- lsreader\_c-vc100-0\_1.dll

| | |-- lsreader\_c-vc100-0\_1.lib

| | `-- lsreader\_c-vc100-s-0\_1.lib

| |-- vs2015

| | |-- lsreader\_c-vc140-0\_1.dll

| | |-- lsreader\_c-vc140-0\_1.lib

| | `-- lsreader\_c-vc140-s-0\_1.lib

| |-- vs2017

| | |-- lsreader\_c-vc141-0\_1.dll

| | |-- lsreader\_c-vc141-0\_1.lib

| | `-- lsreader\_c-vc141-s-0\_1.lib

| `-- vs2019

| |-- lsreader\_c-vc142-0\_1.dll

| |-- lsreader\_c-vc142-0\_1.lib

| `-- lsreader\_c-vc142-s-0\_1.lib

| `-- vs2022

| |-- lsreader\_c-vc143-0\_1.dll

| |-- lsreader\_c-vc143-0\_1.lib

| `-- lsreader\_c-vc143-s-0\_1.lib

|-- lsreadercxxdist

| |-- CMakeLists.txt

| |-- binoutreader.h

| |-- d3plotreader.h

| |-- example.cpp

| `-- lib

| |-- linux

| | |-- liblsreader\_cxx.so -> liblsreader\_cxx.so.0.1.0

| | `-- liblsreader\_cxx.so.0.1.0

| |-- vs2010

| | |-- lsreader\_cxx-vc100-0\_1.dll

| | |-- lsreader\_cxx-vc100-0\_1.lib

| | `-- lsreader\_cxx-vc100-s-0\_1.lib

| |-- vs2015

| | |-- lsreader\_cxx-vc140-0\_1.dll

| | |-- lsreader\_cxx-vc140-s-0\_1.lib

| | `-- lsreader\_cxx-vc140-0\_1.lib

| |-- vs2017

| | |-- lsreader\_cxx-vc141-0\_1.dll

| | |-- lsreader\_cxx-vc141-0\_1.lib

| | `-- lsreader\_cxx-vc141-s-0\_1.lib

| `-- vs2019

| |-- lsreader\_cxx-vc142-0\_1.dll

| |-- lsreader\_cxx-vc142-s-0\_1.lib

| `-- lsreader\_cxx-vc142-0\_1.lib

| `-- vs2022

| |-- lsreader\_cxx-vc143-0\_1.dll

| |-- lsreader\_cxx-vc143-s-0\_1.lib

| `-- lsreader\_cxx-vc143-0\_1.lib

`-- lsreaderwrapperpythondist

|-- example.py

|-- lib

| |-- linux

| | `-- lsreader.so

| `-- windows

| |-- cp36

| | |-- lsreader.pyd

| | `-- msvcp140.dll

| `-- cp37

| |-- lsreader.pyd

| `-- msvcp140.dll

| |-- cp38

| | |-- lsreader.pyd

| | `-- msvcp140.dll

| |-- cp39

| | |-- lsreader.pyd

| | `-- msvcp140.dll

| |-- cp310

| | |-- lsreader.pyd

| | `-- msvcp140.dll

| |-- cp311

| | |-- lsreader.pyd

| | `-- msvcp140.dll

1. **Building and running examples**

We use CMake to build solutions of examples for C++ and C programs on Windows and on Linux.

For Python programs in lsreaderdist, we can directly use command “**python example.py**” to run it. Because we have set the path of lsreader.pyd or lsreader.so in example.py. But if you want to run other Python scripts about lsreader, it is recommended to install lsreader by using “**pip install lsreader**” firstly, more detail see below.

More introductions about APIs of lsreader and how to use, see lsreaderdist/documents:

*LSReader\_C.doc*

*LSReader\_CXX.doc*

*LSReader\_Python.doc*

* 1. **Windows (C++ and C)**

Using CMake to build the solution and project files.

Download CMake from: <https://cmake.org/download/>

You need to use CMake( version >=2.6) in order to build test example for C++ and C

1. Define the source code path(your lsreaderdist folder path) and the build path

A screenshot of a social media post

Description automatically generated

1. Click Configure. If build directory does not exist, a message dialog will pop up. Click Yes.

A screenshot of a social media post

Description automatically generated

1. Setup the proper Visual Studio version and the platform choice, then click “Finish”

A screenshot of a social media post

Description automatically generated

1. Wait until Cmake finish, then click “Generate” to finish up the Cmake operations

A screenshot of a social media post

Description automatically generated

1. cd to the build folder, select lsreaderdist.sln, right click on it to “open with” your Visual Studio version

A screenshot of a cell phone

Description automatically generated

1. Select “release” or “debug”, and right click “ALL\_BUILD” and select “Build”, make sure there are 0 failed at the end of the build

A screenshot of a computer screen

Description automatically generated

1. cd to \build\_lsreader\bin\lsreadcdist\Release, and double click “cmain.exe”

A screenshot of a cell phone

Description automatically generated

* 1. **Linux (C++ and C)**

You also need to use CMake(>=2.6) to build solutions of examples.

\* Make a new directory (such as "build\_lsreader") to receive cmake files and bin folder

Type:

**cd build\_lsreader**

**cmake your/lsreaderdist/path**

**make**

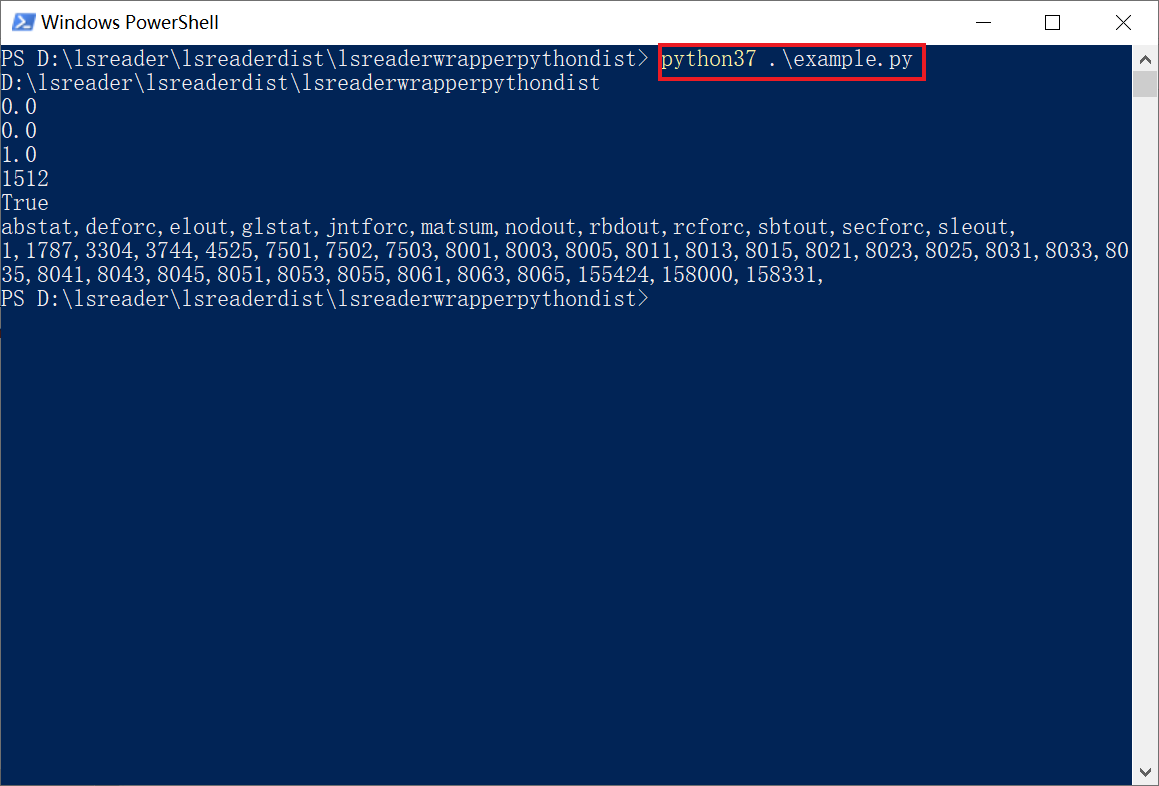
* 1. **Python (Windows and Linux)**

As we said before, running example.py in lsreaderdist is very simple on Windows and on Linux. We just need to use:

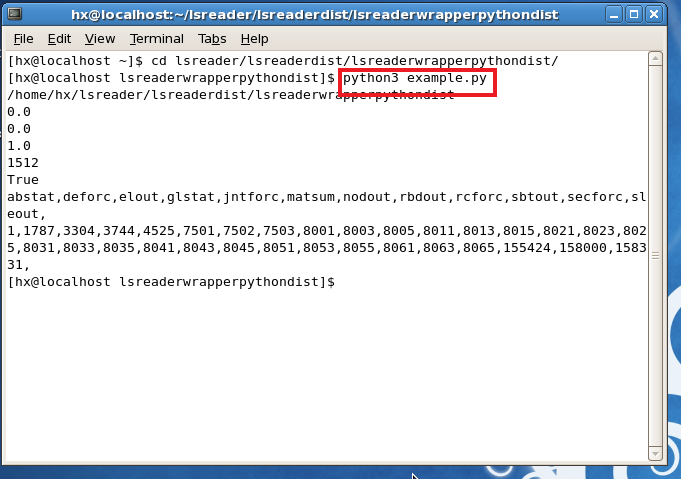
**python example.py**

in lsreaderdist/lsreaderwrapperpythondist/.

Windows:



Linux:



If you want to use LS-Reader for Python. It is recommended to install lsreader by using pip command.

pip is the package installer for Python. You can use pip to install packages from the Python Package Index and other indexes. The usage of pip is very simple:

**pip install lsreader –-user**

If some errors occur when installing lsreader by using pip, please update pip first and then use pip command to install lsreader:

**pip install --upgrade pip --user**

**pip install lsreader –user**

If you have multiple Python versions installed on your system (e.g. Python3.6 and Python3.7), you can use command:

**Python3.n -m pip install –-upgrade pip --user**

**Python3.n -m pip install lsreader --user**

to specify which version will be used. Python3.n is an executable for your different Python, and must have been added to the environment variables.

More details about lsreader on the Python Package Index(PyPI), see <https://pypi.org/project/lsreader/>

1. **Adding programs in lsreaderdist**

If you want to add own C/C++ programs in lsreaderdist, you should write the corresponding CMakeLists.txt. If you want to add Python codes, it is recommended to install lsreader by using pip. Otherwise, you must specify the path of lsreader.pyd or lsreader.so in your Python programs(Of course, you also can put lsreader.pyd or lsreader.so and your Python programs into the same directory, and then Python will find these libraries).

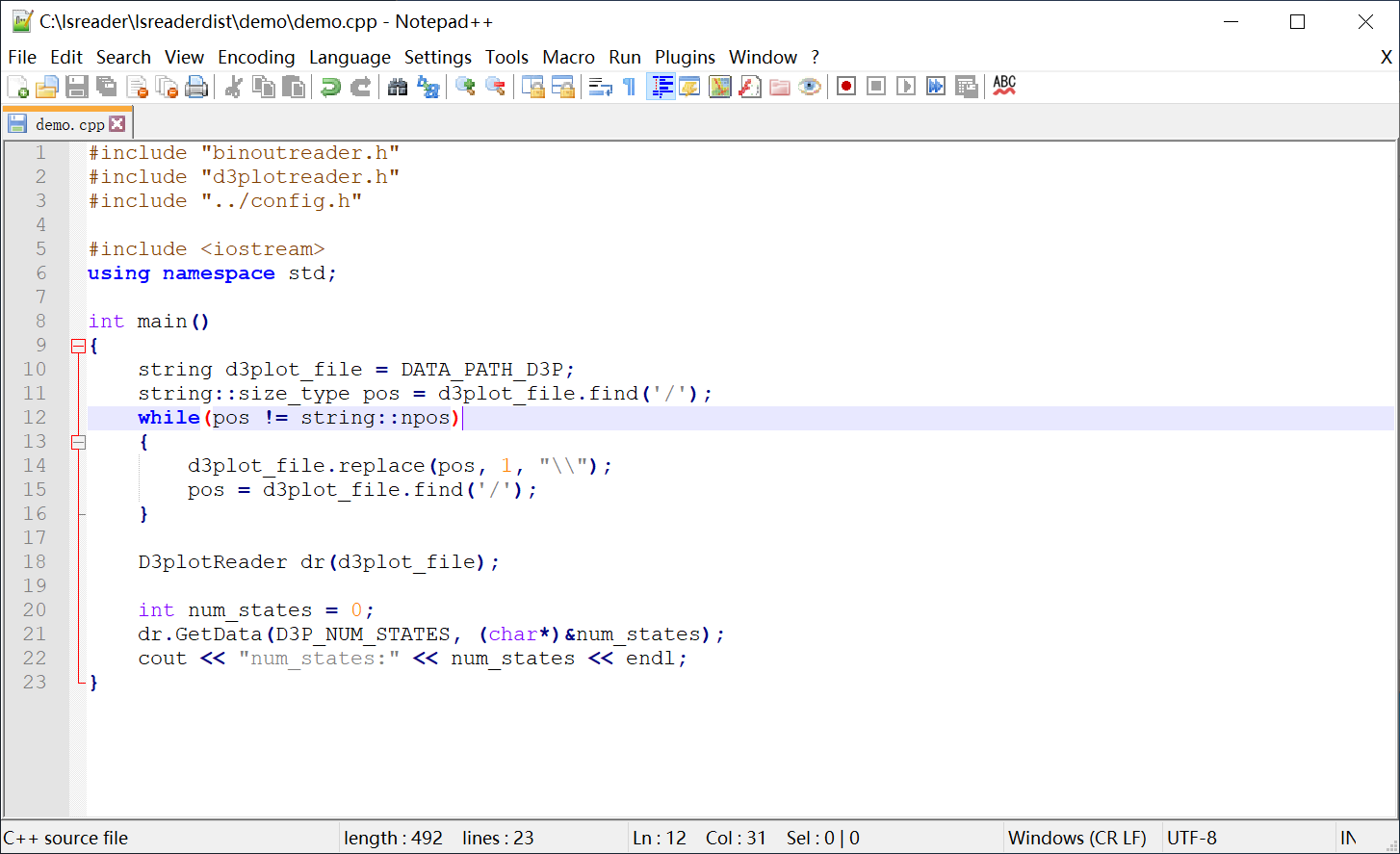
These are steps about how to add your C/C++ programs in lsreaderdist:

1. Make a new directory to store your codes.

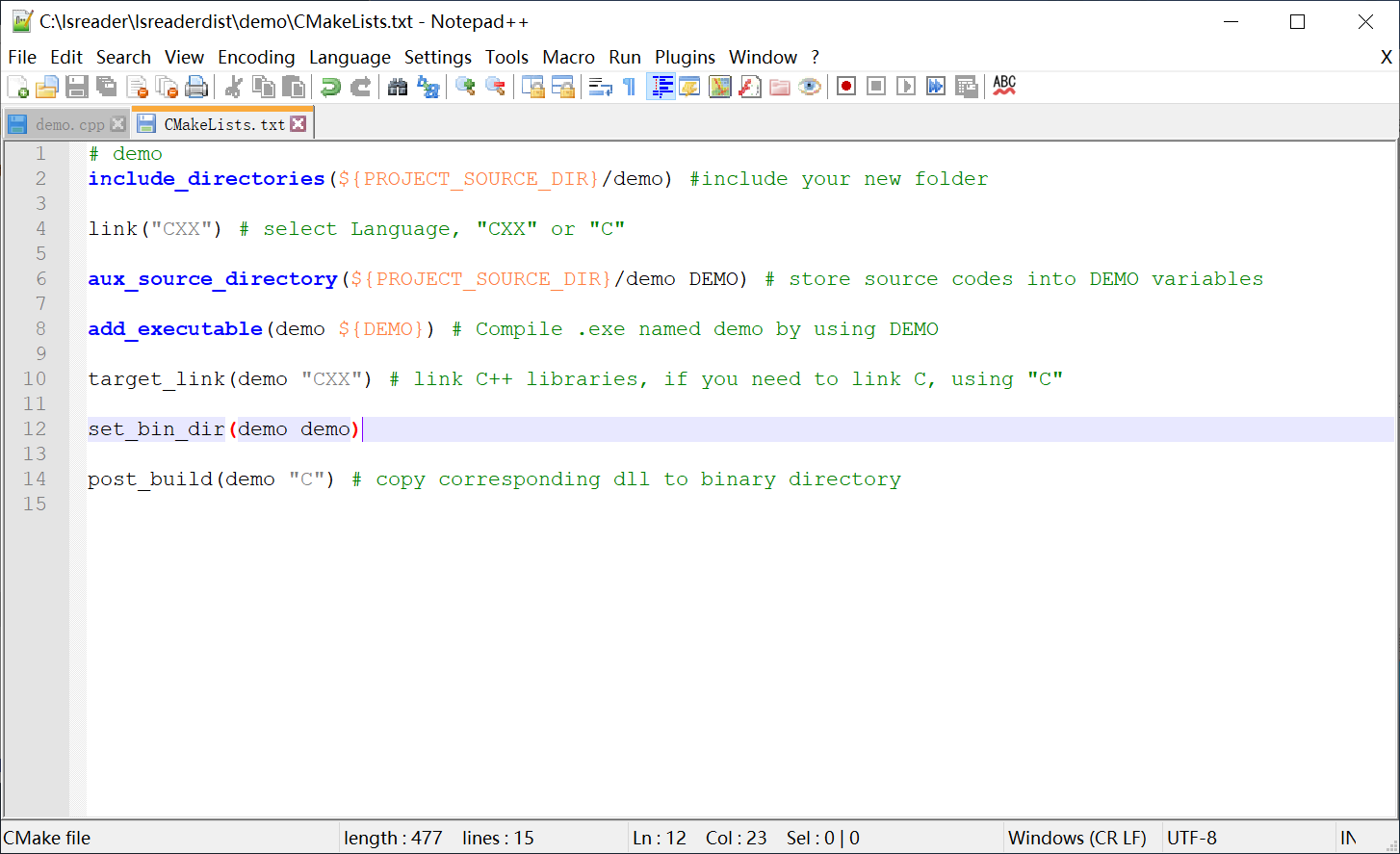
Graphical user interface, text, application

Description automatically generated

1. Write your codes.



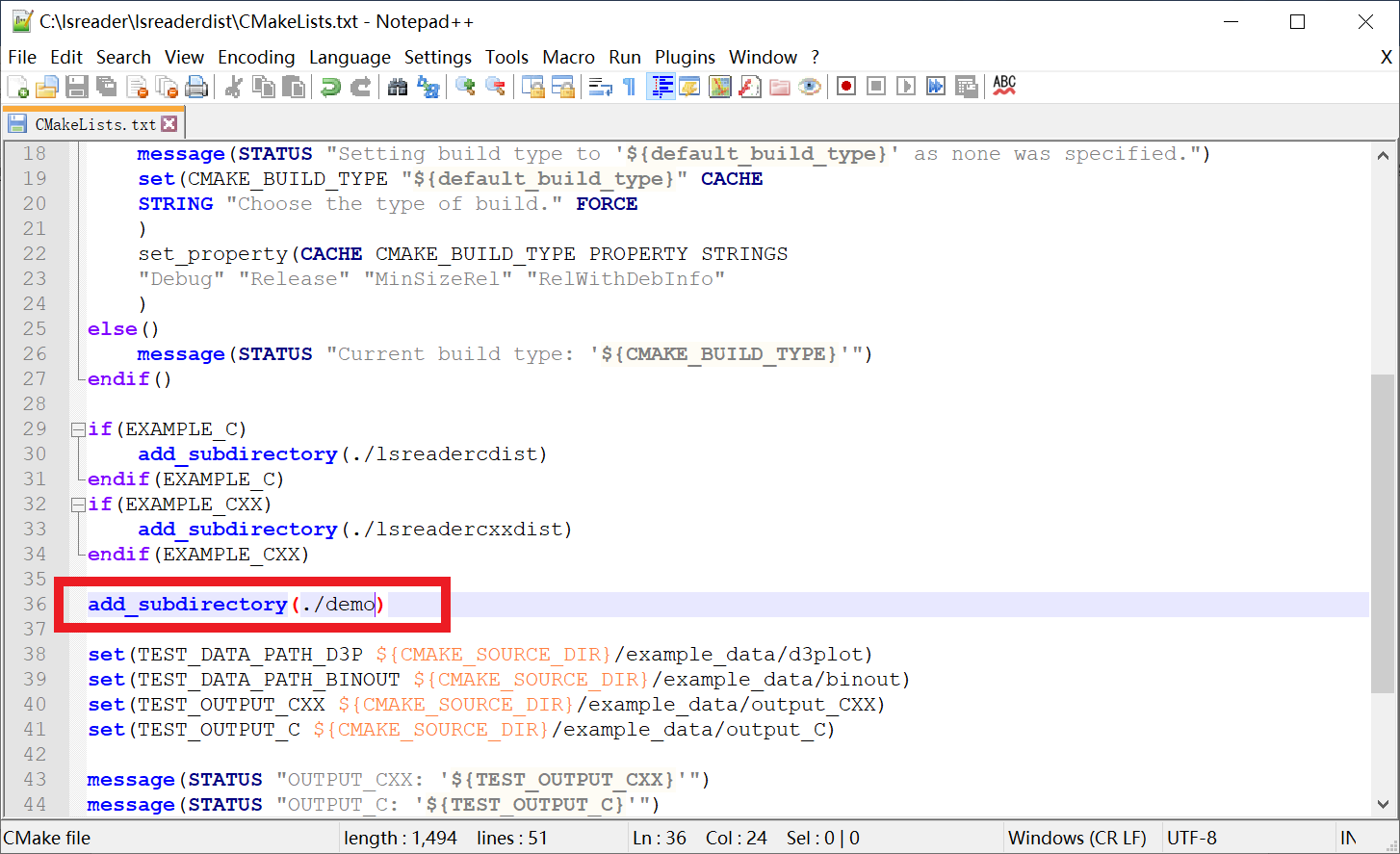
1. Write CMakeLists.txt by using “Macros.cmake”, and put it into your folder.



Graphical user interface, text, application

Description automatically generated

1. Modify the CMakeList.txt in root directory of lsreaderdist.



1. Compile and build it in the same steps as **Building and running examples.**