How to install Linux Wrapper for Coherence5LE.dll

The Wrapper needs Wine (http://www.winehq.com) to work. Therefore you need to install Wine and the Wine development package. Either download from the winehq.com or use the package managing software coming with your distribution (apt-get, portage, rpm, ...)
For example for a Debian-based Linux distributions install the two packages *wine* and *wine-dev* with:

```
sudo apt-get install wine wine-dev
```

After installing wine, the wrapper can be compiled with

```
./makeWrapper.sh
```

Make sure that makeWrapper.sh is executable by setting chmod 755
After this you should be able to use the dll as a shared object in Linux.

Example Files

testwrapper.c	Example File how to use the shared object
testdll.c	Example File to test dll-functions with wine
testCoherence.m	Example for using Coherence5LE.dll in Matlab

Linux Wrapper Functions

For the use of the structs either have a look into Coherence5LEstructs.h or the Coherence5LE.dll Documentation.

If not stated otherwise the functions work like described in the Coherence5LE.dll Documentation.

int Eeg3_Initialisation()

int Eeg3_Termination()

int Eeg3_Unlock(TUnlock3LE unlockstr)

int Eeg3_GetEeg(int begin, int duration, short* buffer)
see Eeg3_GetEeg2(int,int,short*) from dll

int Eeg3_OpenFile(char* filename, Tcoh3* fileinfo)

int Eeg3_CloseFile()

int Eeg3_Version(TVersion* version)

int Eeg3_GetMarkers(int begin, int end, Tmarker* markers)
see Eeg3_GetMarkers2(int,int,Tmarkers*) from dll

int Eeg3_PutMarker(TMarker* marker)

int Eeg3_GetImpedances(int startpos, Timpedances* impedances)

int Eeg3_NextFile(int direction, char* newfilename, Tcoh3* fileinfo)

int Eeg3_DebugFileSwitch(bool onoff)

int Eeg3_GetMarkersNumber(int begin, int end)

MatlabFunctions

To use the Dll/Shared Object in Matlab you can use the Coherence5LE Object. For example usage see testCoherence.m. Place your unlock code in @Coherence5LE/unlock.m.

z = Coherence5LE(unlockcode)
Creates a Coherence5LE Object. Unlockcode is optional, if not supplied the hardcoded unlockcode from unlock.m is used.

z = display(z)
returns String representation of z

data = GetData(z)
returns all EEG-Data from z. Data is an array with size *samples x electrodes*

[data, errorcode] = GetEeg(z, start, duration)
returns next *duration* samples from eeg-data beginning at *start*. Errorcode should be the number of samples read (*duration* in most cases) and -106 on end of file.

FileInfo = GetFileInfo(z)
return information about the currentfile. FileInfo is a struct similar to TCoh3

impedances = GetImpedances(z, startpos)
returns impedances as a struct similar to Timpedances

markers = GetMarkers(z, beginpos, endpos)
returns Markers in the interval [beginpos endpos] as array of Tmarker structs. If beginpos and endpos are not specified, markers from the whole file are returned.

number = GetMarkersNumber(z, beginpos, endpos)
returns the number of markers in the interval [beginpos endpos]. If beginpos and endpos are not specified, number of markers in the whole file is returned.

[z,res] = NextFile(z, direction)
load the nextfile when using bloc-files. Use *direction* 1 for next and -1 for previous. *res* is the errorcode and should be 0 if no error occurred. -1208 if the file is no bloc-file and -10 if there is no more file in the given direction.

z = OpenFile(z, filename)
Opens the file specified by filename

[z, errorcode] = PutMarker(z,pos, duration, text , eventtype)
Puts a Marker at given position. Errorcode should be 0 if no error occurred and 3 if this marker already has been written. -2005 means that file is not writeable.

z = unlock(z, unlockcode)
can be used to manually unlock the dll. If the unlockcode is hardcoded in this file, there will be no need to unlock manually.