**DirectX**

**Creazione di filtri per DirectShow in C++**

**(Fischetti P.)**

Scaricare Windows-classic-samples-main da [1] e decomprimere

Posizionarsi con il command prompt di Visual studio nella relativa cartella ad esempio:

C:\Windows-classic-samples-main\Samples\Win7Samples\multimedia\directshow\baseclasses

Impostiamo gli include e i lib

|  |
| --- |
| SET INCLUDE=%INCLUDE%; C:\Windows-classic-samples-main\Samples\Win7Samples\multimedia\directshow\baseclasses  SET LIB=%LIB%; C:\Windows-classic-samples-main\Samples\Win7Samples\multimedia\directshow\baseclasses |

Creiamo un file ad es. buildLib.bat contenente tutti i file cpp presenti nella cartella:

|  |
| --- |
| BuildLib.bat |
| cl -c /EHsc /DUNICODE /D\_UNICODE amextra.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE amfilter.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE amvideo.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE arithutil.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE combase.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE cprop.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE ctlutil.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE ddmm.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE dllentry.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE dllsetup.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE mtype.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE outputq.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE perflog.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE pstream.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE pullpin.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE refclock.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE renbase.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE schedule.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE seekpt.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE source.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE strmctl.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE sysclock.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE transfrm.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE transip.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE videoctl.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE vtrans.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE winctrl.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE winutil.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE wxdebug.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE wxlist.cpp  cl -c /EHsc /DUNICODE /D\_UNICODE wxutil.cpp  lib /OUT:strmbase.lib @libInFile.txt |

Creiamo un file ad esempio libInFile.txt contenente gli stessi nomi precedenti ma con estensione obj

|  |
| --- |
| libInFile.txt |
| amextra.obj  amfilter.obj  amvideo.obj  arithutil.obj  combase.obj  cprop.obj  ctlutil.obj  ddmm.obj  dllentry.obj  dllsetup.obj  mtype.obj  outputq.obj  perflog.obj  pstream.obj  pullpin.obj  refclock.obj  renbase.obj  schedule.obj  seekpt.obj  source.obj  strmctl.obj  sysclock.obj  transfrm.obj  transip.obj  videoctl.obj  vtrans.obj  winctrl.obj  winutil.obj  wxdebug.obj  wxlist.obj  wxutil.obj |

Ora dopo aver aperto il Command prompt di visual studio :

lanciare il precedente comando BuildLib.bat che in mancanza di errori produrra il file strmbase.lib

Costruiamo un filtro DirectShow sfruttando quelli presenti nel file Git precedentemente scaricato. L’esempio ezrgb24 crea un filtro per impostare degli effetti grafici nei filmati:

cl -c /EHsc /Zc:wchar\_t- ezrgb24.cpp

cl -c /EHsc /Zc:wchar\_t- ezprop.cpp

RC ezprop.rc

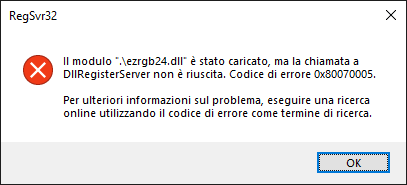
link -nologo -dll /OUT:ezrgb24.dll /NODEFAULTLIB:msvcrt.lib /NODEFAULTLIB:libcmt.lib /def:ezrgb24.def ezrgb24.obj ezprop.OBJ ezprop.res ..\strmbase.lib strmiids.lib quartz.lib vfw32.lib winmm.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib msvcrtd.lib

se tutto ok viene creato il file ezrgb24.dll (uin alcuni esempi al posto di .dll si uo’ trovare l’estensione .ax cioe’ ActiveX ma e’ sempre una dll).

Ora occorre registrare la dll appena crata lanciare il prompt del DOS e digitare:

C:\>regsvr32 .\ezrgb24.dll

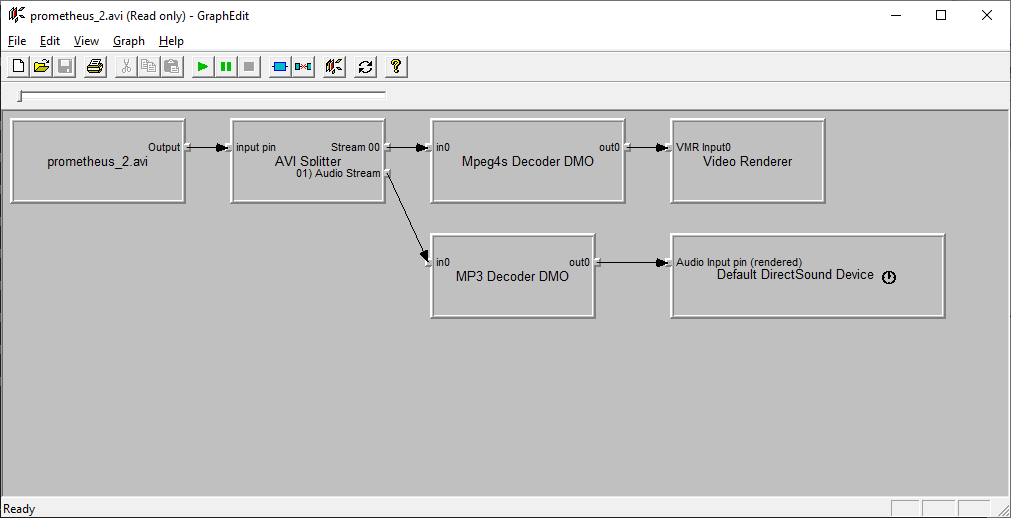
N.B. se si ottiene il seguente messaggio relativo all’errore di sicurezza 0x80070005:



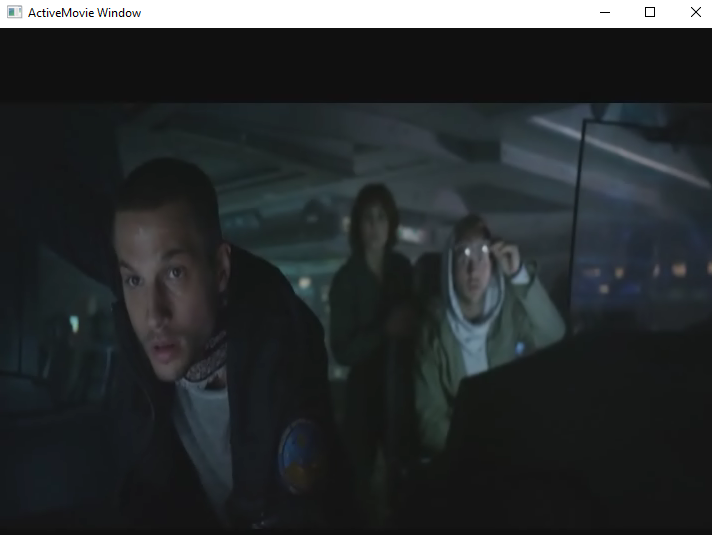
Lanciare il prompt come administrator.

Se la dll viene registrata con successo potremo vederla nel passaggio seguente:

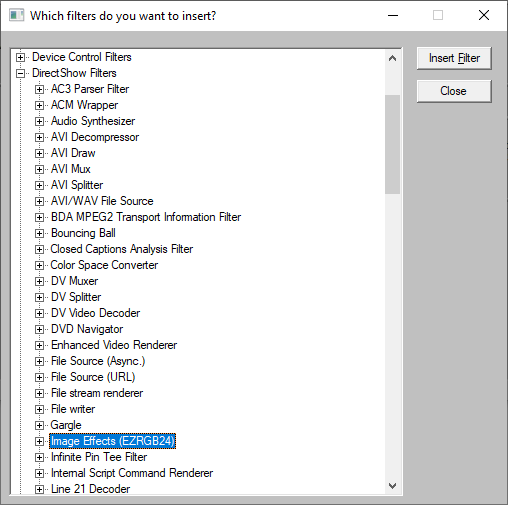
Lanciare il programma Microsoft Graphedt.exe [2] e caricare un file video



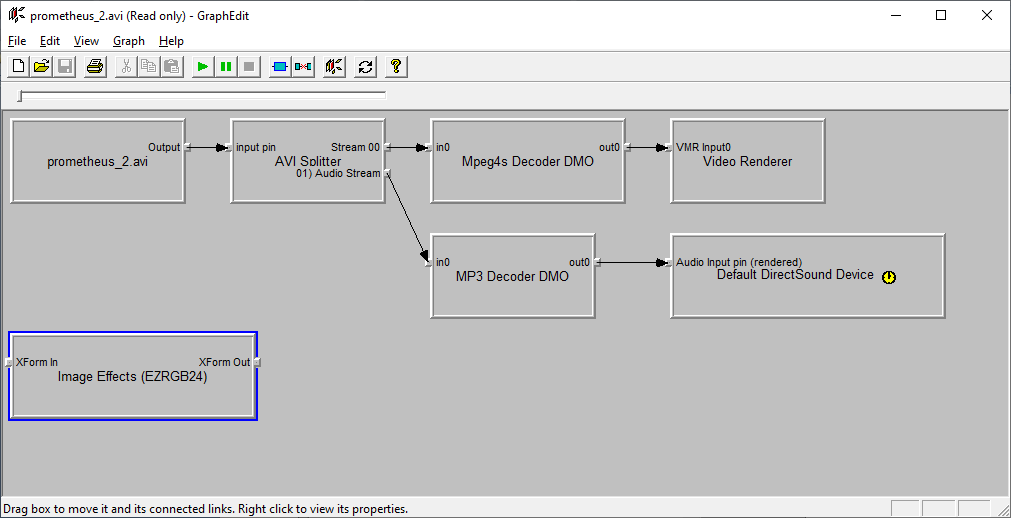
Play:



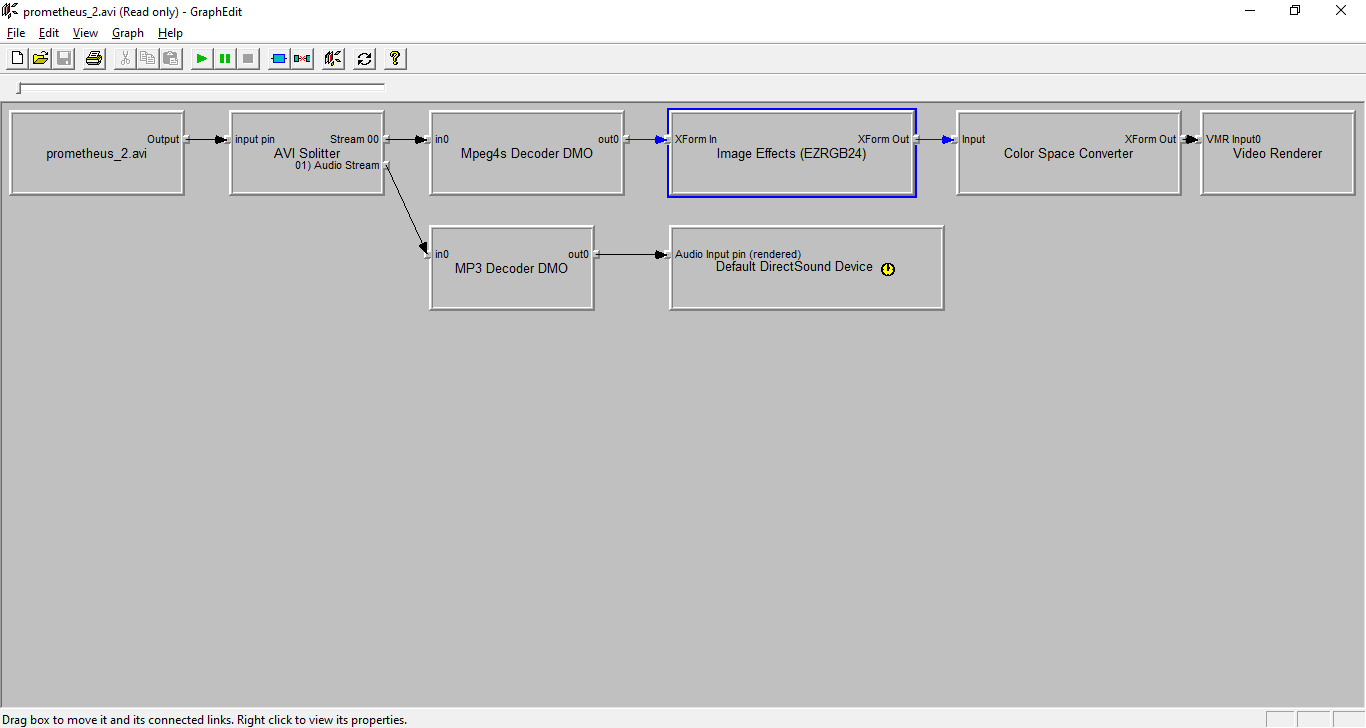
Chiudere il video e cliccare su menu: Graph->Insert Filters…



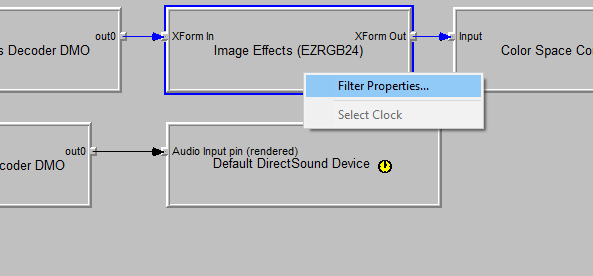
Apparira’ il nostro nuovo filtro (ovviamente se precedentemente la registrazione e’ andata a buon fine), inseriamolo:



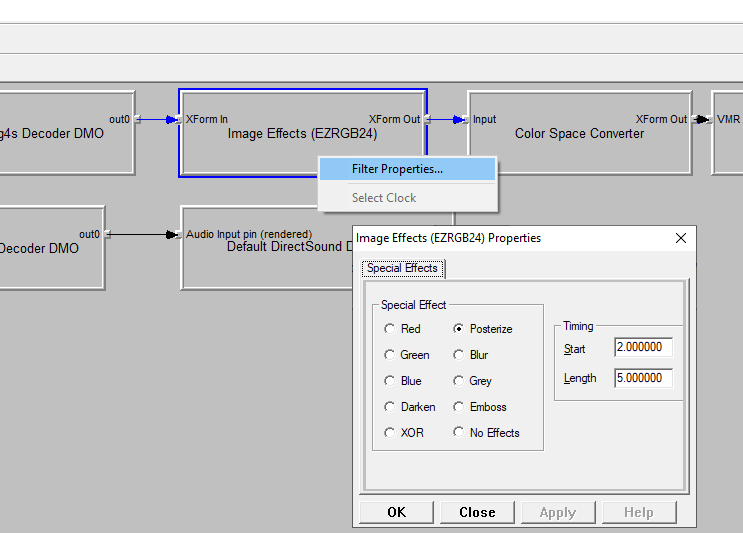
agganciare il nostro filtro prima del blocco ‘Video Render’:



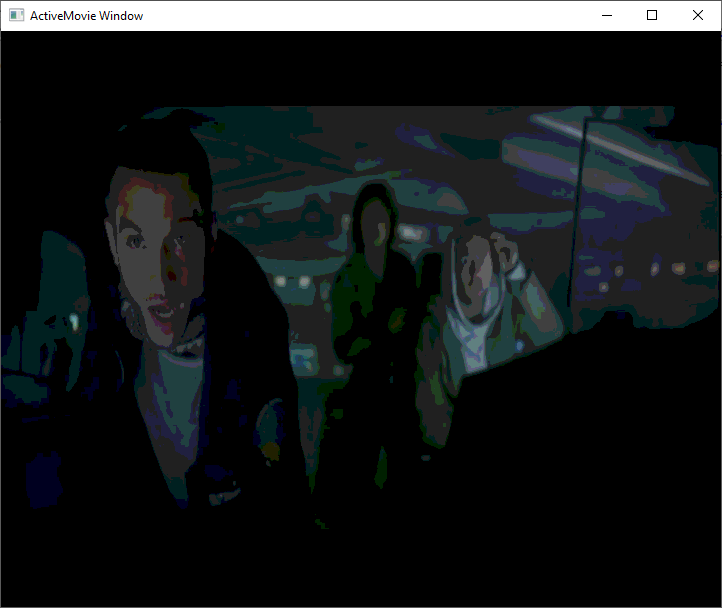
Clic destro sul nostro filtro:

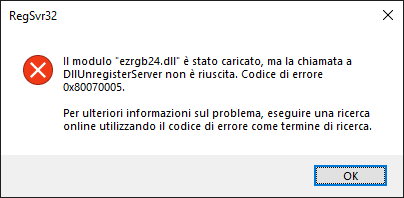


Scegliamo ad esempio l’effetto posterize poi OK e riavviamo il filmato:



Si dovrebbe ottenere il filmato coprecedente con l’effetto ‘posterize’:



Ovviamente con il comando da DOS (come administror per evitare):

C:\> regsvr32 /u ezrgb24.dll

Il filtro non sarebbe piu’ registrato nel registro di sistema cioe’ non lo vedremo piu’ nell’elenco dei filtri in ‘Direct Show’ come prima.

N.B.!: Attenzione alla coerenza x86 e x64: cioe’ se compilo le librerie e i filtri come x86 dovro’ usare la versione x86 di Graphedt.exe, altrimenti la x64 pena l’assenza del filtro nell’elenco ‘DirectShow Filters’, anche se il filtro e’ stato correttamente registrato

Riferimenti:

[1] <https://github.com/microsoft/Windows-classic-samples>

[2] C:\Program Files (x86)\Windows Kits\10\bin\10.0.23481.0\x64