

Visualization Toolkit (VTK)

Manual de Instalación

Proyecto de Servicio Social Ene-Dic 2019

Roberto Cordero Salas & Alfonso Zumano Rivera (UAQ – Informática)

Introducción

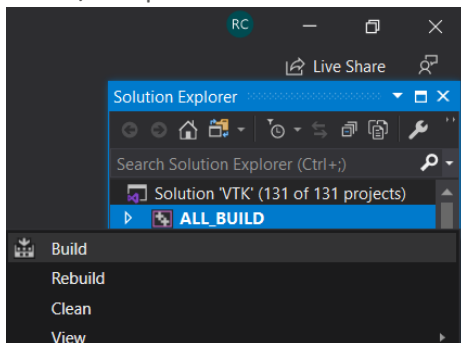
El Kit de herramientas de visualización (VTK) es un sistema de software libre, disponible para la realización de gráficos 3D por computadora, procesamiento de imagen y visualización. VTK consiste en una biblioteca de clases de C++ y varias capas de interfaz interpretadas como Tcl/Tk, Java, y Python.

Instalación general (Windows)

1. Descargar y descomprimir la última versión de VTK – [clic aquí](#).
2. Descargar la última versión de CMake – [clic aquí](#).
3. Crear un folder para la versión compilada de VTK.
4. Ejecutar CMake e insertar las URLs correspondientes:

```
Where is the source code: C:\MyProjects\VTK-src  
Where to build the binaries: C:\MyProjects\VTK-bin
```

5. Clic en Configure y seleccionar los elementos requeridos para nuestro proyecto.
6. Posteriormente volver a dar clic en Configure para verificar que todo este como lo queremos (no debe quedar nada en color rojo).
7. Dar clic en Generate y esperar a que termine. Una vez finalizado podemos cerrar CMake.
8. Ejecutar Visual Studio como administrador y abrir la solución (archivo .sln) que se encuentra en el folder creado en el paso 3.
9. En Visual Studio seleccionar si se quiere realizar la compilacion de VTK en la configuración RELEASE o DEBUG.
10. En el panel de la derecha dar clic derecho sobre ALL-BUILD y después en BUILD/Compilar



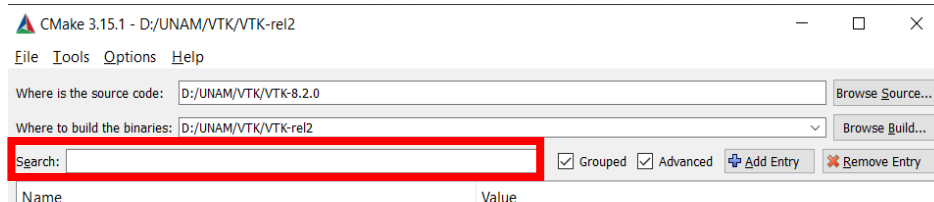
11. Cuando termine de compilarse, dar clic derecho a INSTALL (se encuentra en la panel derecho) y posteriormente en BUILD/Compilar.
12. Esperar a que se termine y cerrar Visual Studio.

Ya podemos empezar hacer proyectos siguiendo las pautas establecidas en los ejemplos de VTK. [Clic aquí para ejemplos](#).

Instalación para LAVIS Visualizer (Windows)

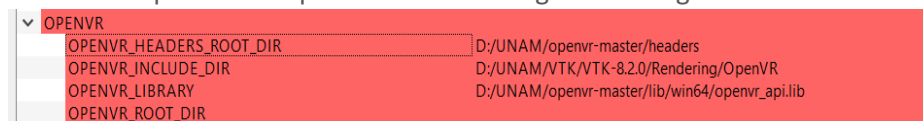
1. Realizar los pasos 1 a 4 del proceso de instalación general de VTK.
2. Descargar y descomprimir el SDK de OpenVR proporcionado por Valve. [Link aquí](#).
3. En CMake, dar clic en configure y seleccionar los siguientes elementos:
 - a. BUILD_SHARED_LIBS
 - b. Module_vtkRenderingOpenVR

Si no aparecen, se pueden buscar en la barra de búsqueda de CMake.



4. Dar clic en 'Configure' y esperar a que se configure VTK. Aparecerá un mensaje de error mencionando que no se encontró OpenVR.
5. Ya sea que se mostro como 'Ungrouped Entries' o 'OPENVR' desglosamos ese elemento y llenamos lo que se pide con los siguientes datos:
 - a. OPENVR_HEADERS_ROOT_DIR → carpeta donde se encuentra los encabezados de OpenVR, estos los obtenemos de la carpeta que descomprimos en el paso 2, en la subcarpeta 'headers'.
 - b. OPENVR_INCLUDE_DIR → esta opción se debió llenar automáticamente, en caso contrario insertar la URL donde tenemos el código fuente de VTK (la 1ra url que insertamos en CMake) y a sus subcarpetas '/Rendering/OpenVR'.
 - c. OPENVR_LIBRARY → ponemos la url donde se encuentra el archivo *.lib de OpenVR, que se encuentra en la carpeta descomprimida en el paso 2, en la subcarpeta '/lib/win64/openvr_api.lib'. La carpeta de win64 puede variar a win32 si su sistema operativo es de 32 bits (x32 o x86 bits).

Al final esta parte debe quedar similar a la siguiente imagen:

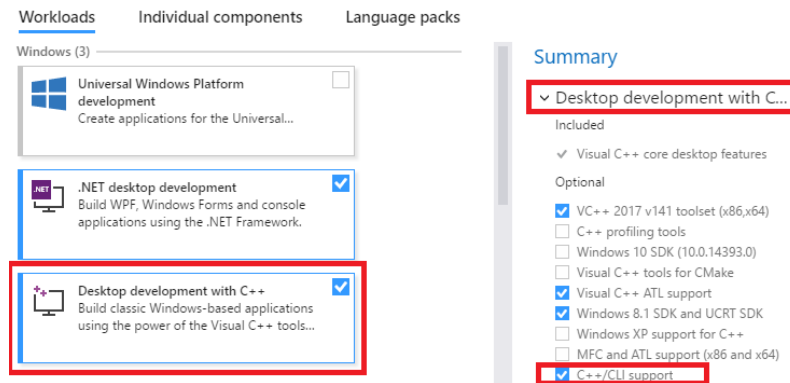


6. Volvemos a dar clic en 'Configure' y al finalizar debería aparecer algo como 'OpenVR found'.
7. Posteriormente damos clic en 'Generate' y realizamos los pasos 8 a 12 del proceso de instalación general.

8. Verificamos que el Visual Studio que tenemos (recomendado la versión mas reciente) tenga habilitada la opción de 'CLR Empty Project'.

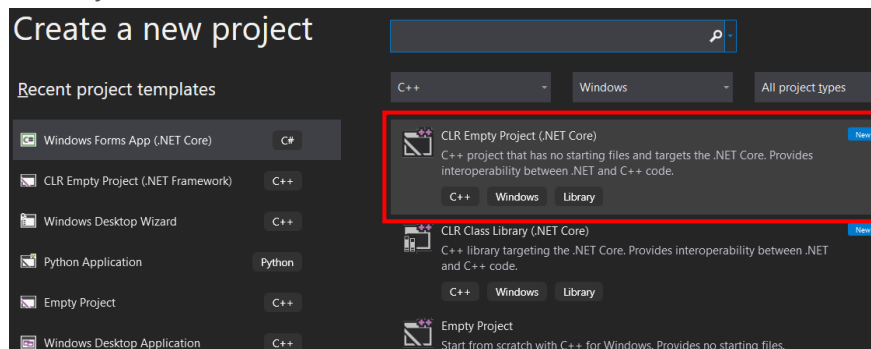
a. En caso de no estar habilitada esta opción realizar lo siguiente:

i. Ejecutar Visual Studio Installer y en la categoría de 'Desktop development with C++' seleccionar el elemento 'C++/CLI Support'.



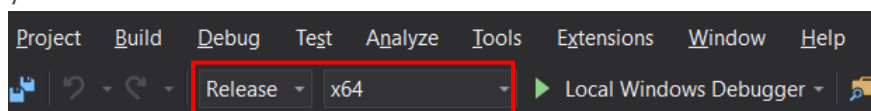
b. Descargamos e instalamos esta característica. Si pide reiniciar el dispositivo damos clic en reiniciar o reiniciamos manualmente.

c. Para verificar la instalación ejecutamos Visual Studio y damos clic en 'Crear nuevo proyecto' y verificamos que exista la opción 'CLR Empty Project'.

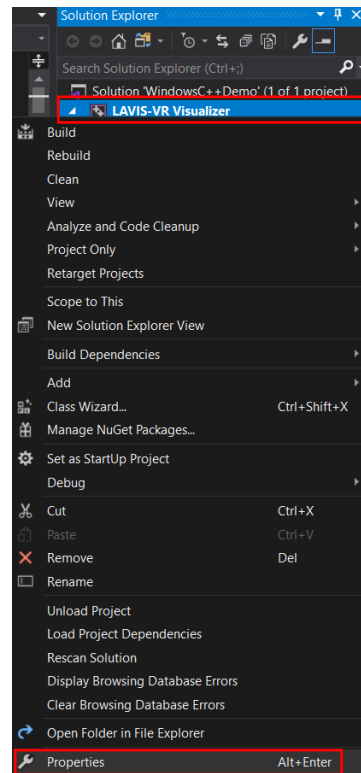


9. En la carpeta de LAVIS Visualizer del escritorio hay una subcarpeta llamada 'WindowsC++Demo' dentro de esta carpeta se encuentra un archivo llamado 'WindowsC++Demo.sln' ejecutarlo.

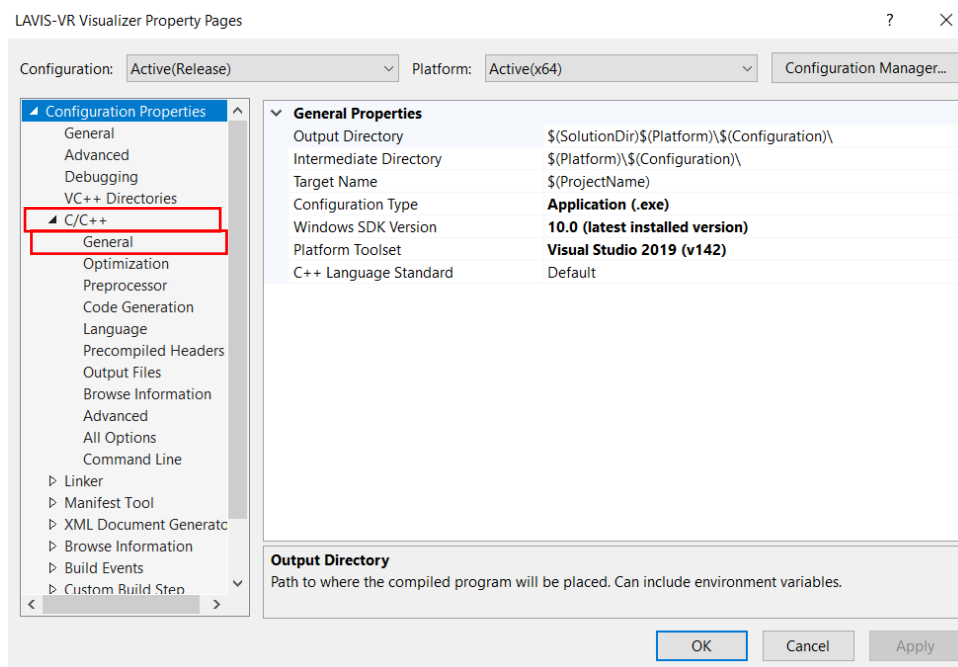
10. Verificamos que en la parte superior de Visual Studio su configuración se 'Release' y 'x64'.



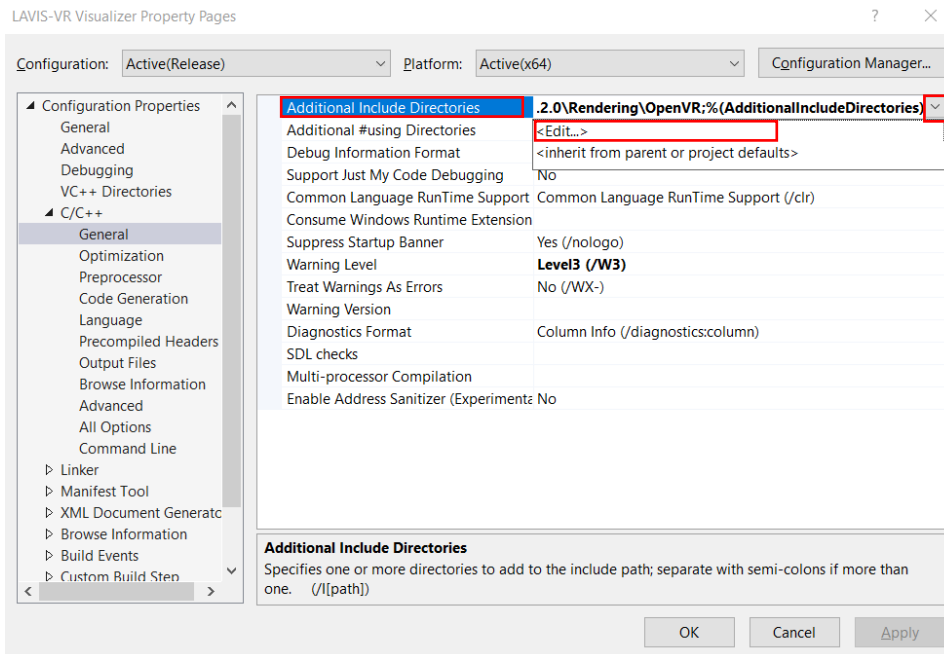
11. Ir al panel de la derecha (Solution Explorer) y dar clic derecho en 'LAVIS VR-Visualizer' y posteriormente en 'Propiedades'. Como se muestra en la imagen de la derecha →.



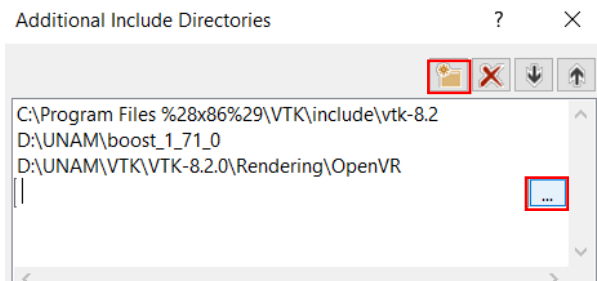
12. Aparecerá el siguiente cuadro de dialogo en el cual nos dirigimos a la sección de 'C++' y posteriormente a 'General'.



13. Ya estando en esa sección damos nos dirigimos a la parte de 'Additional Include Directories', al final de la parte blanca o del renglón aparecerá una flecha para desglosar un menú, damos clic y después clic en 'Edit'.



14. Aparecerá un nuevo cuadro de dialogo, damos clic en el icono de nueva línea (esta a la izquierda de una tache/x), aparecerá una nueva línea en blanco y un botón con puntos suspensivos a la derecha, damos clic en ese botón.

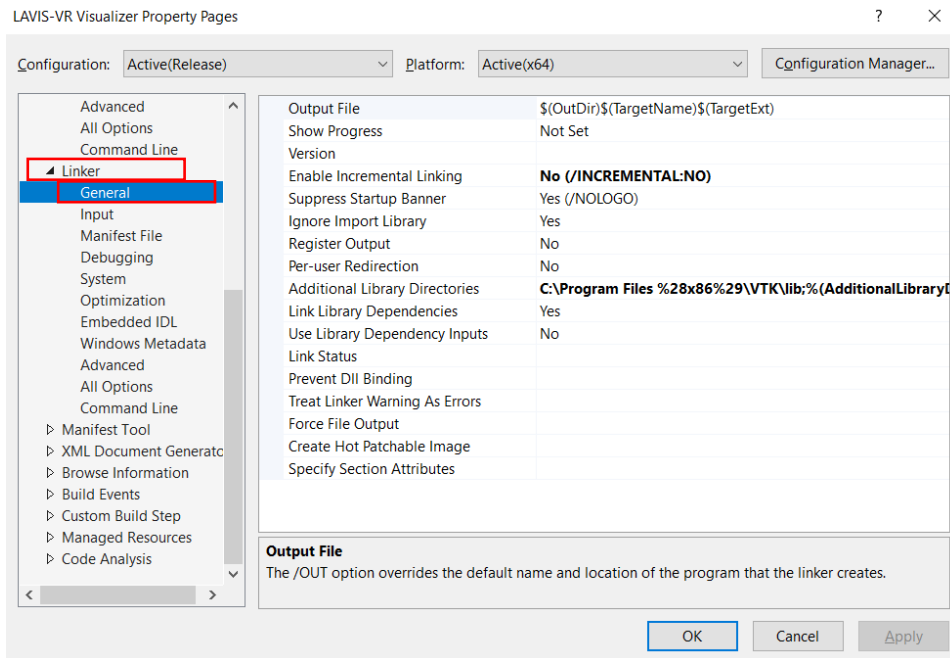


15. Ingresamos las direcciones/urls de las siguientes carpetas:
- Nos dirigimos al disco local C → Program Files x86 → VTK → include → vtk-8.2. Seleccionamos ese folder (clic en la parte inferior derecha del cuadro de dialogo).
 - Repetimos paso 13 y seleccionamos la carpeta de la librería boost ubicada en la carpeta del proyecto LAVIS Visualizer → VTK → boost_1_71_0. Seleccionamos esa carpeta.
 - Repetimos paso 13 y seleccionamos la carpeta ubicada en LAVIS Visualizer → VTK → VTK-src(8.2) → Rendering → OpenVR. Seleccionamos esa carpeta.

Al final debería quedar algo similar a la imagen superior.

Damos en 'OK' (se cerrara el cuadro de dialogo) y posteriormente en 'Apply'.

16. Posteriormente nos dirigimos a la sección de 'Linker' o 'Vinculador' y después en 'General'.



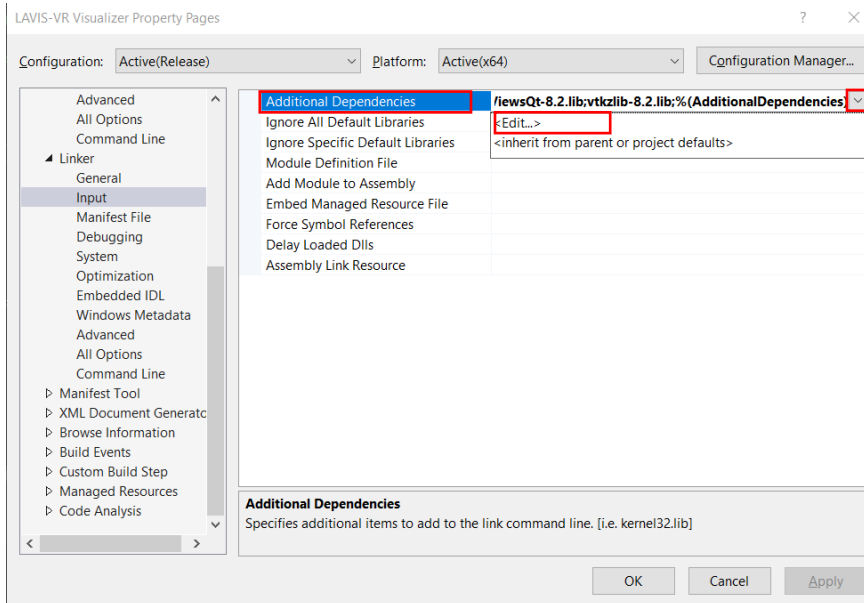
17. En esa sección damos nos dirigimos a la parte de 'Additional Include Directories', al final de la parte blanca o del renglón aparecerá una flecha para desglosar un menú, damos clic y después clic en 'Edit'.

18. Ingresamos la siguiente dirección (como en el paso 13 y 14):

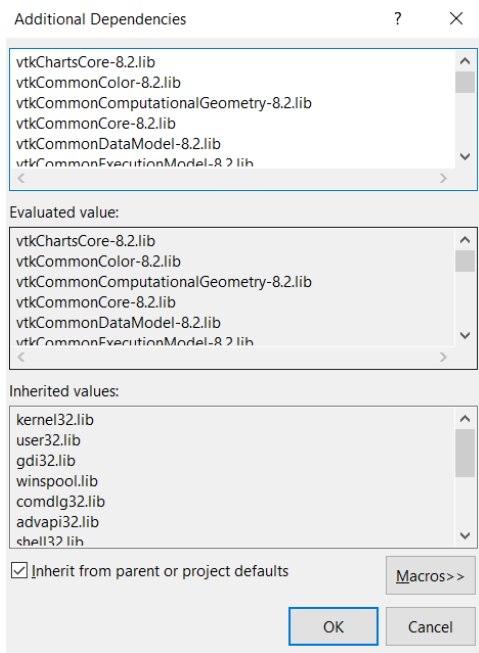
Disco C: → Program Files x86 → VTK → lib

Seleccionamos esta carpeta. Damos clic en 'OK' y posteriormente en 'Apply'.

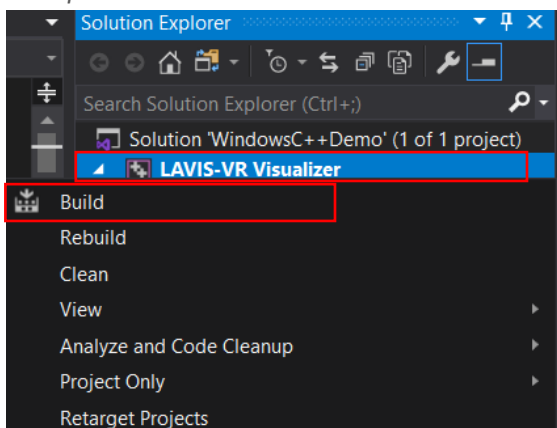
19. En 'Linker' nos dirigimos ahora a la sección de 'Input' y seleccionamos 'Additional Dependencies', al final del renglón aparecerá una flecha para desglosar un menú, damos clic y después clic en 'Edit'.



20. Aparecerá en pantalla el siguiente cuadro de dialogo.



21. Para este cuadro tendremos que abrir el archivo de texto '*vtkLibs*' que se encuentra en la carpeta del proyecto LAVIS Visualizer → LAVIS Visualizer → Docs.
22. Seleccionamos y copiamos todo el contenido del archivo y lo pegamos en el cuadro de dialogo anterior en el recuadro superior.
23. Damos clic en '*OK*', luego en '*Apply*' y finalmente en '*OK*'. Se cerrara el cuadro de dialogo de las propiedades del proyecto.
24. Para finalizar nos volvemos a dirigir al panel de la derecha (Solution Explorer), damos clic derecho en '*LAVIS VR-Visualizer*' y posteriormente en '*Build*' o '*Compilar*'.



Esperamos a que se termine de compilar el proyecto.

25. Podemos cerrar Visual Studio y nos dirigimos a la carpeta del proyecto en LAVIS Visualizer → x64 → Release → LAVIS-VR Visualizer.exe
Este archivo es el ejecutable de nuestro proyecto.