Installing VTK 4.1.x on a windows PC

Revision: 1.8

Rasmus Paulsen
rrp@imm.dtu.dk

August 6, 2002

1 Introduction

This note describes how VTK version 4.1.x can be installed on a windows PC. It is probably not the best or easiest way to do it, but it has worked for me. This guide only shows how VTK is installed with support for `C++` and `TCL/TK`. Building VTK is the only way to get access to the patented algorithms since they are not present in the prebuilt binaries.

For additional info and for a general introduction to VTK, I recommend buying the books [1, 2].

Windows is not case sensitive regarding filenames and directories. Backslashes (\) are only used where I believe they are necessary.

2 Prerequisites

- Visual Studio 6.0 installed
- Unpacker installed (WinZip or equivalent)
- Previous installations of VTK completely uninstalled. Also all dlls from earlier versions of VTK must be removed. Do a complete search for `vtk*.dll` and remove all old vtk dlls.

3 Download source and unpack

Always check the dashboard at `http://public.kitware.com/VTK` to see if there are any serious problems with the nightly release. Wait 24 hours if there is any problems, then they will probably have been solved.

Download `http://public.kitware.com/VTK/files/nightly/vtkNightlySrc.zip`. Unzip it to `C:\` thus creating a directory tree starting with `C:\VTK\`.

Alternatively use CVS to get the source code.

Downlad `http://public.kitware.com/VTK/files/nightly/vtkNightlyData.zip` from the download page. Unzip it to `C:\` thus creating a directory tree starting with `C:\VTKData\`.

Alternatively use CVS to get the data.

Download `http://www.cmake.org/CMSetup.exe` and install it.

Download ActiveTcl from `http://aspn.activestate.com/ASPN/Downloads/ActiveTcl/` and install it in the default location `C:\Tcl\`. As an alternative a light version of TCL/TK can be downloaded from `ftp://public.kitware.com/pub/vtk/misc/tcltk/` and installed to `C:\Tcl\`.

4 Using CMake to generate projects

Start CMake. Enable `Show Advanced Values` and fill out or verify the following fields (most of them should have the correct values already). Due to the rapid change of VTK and CMake some of this fields change over time.

- Where is the source code: `C:\VTK\`
- Where to build the binaries: `C:\vtkbin\`
- VTK_WRAP_TCL_EXE=C:/vtkbin/bin/$(IntDir)/vtkWrapTcl.exe
- VTK_WRAP_HINTS=C:/VTK/Wrapping/hints
- VTK_USE_64BIT_IDS=OFF
- VTK_MANGLE_MESA=OFF
- VTK_DISPLAY_WIN32_WARNINGS=OFF
- VTK_DEBUG_LEAKS=OFF
- TK_XLIB_PATH=C:/Tcl/include
- TK_WISH=C:/Tcl/bin/wish83.exe
- TK_LIBRARY=C:/Tcl/lib/tk83.lib
- TK_INTERNAL_PATH=C:/VTK/Rendering/tkInternals/tk83
- TK_INCLUDE_PATH=C:/Tcl/include
- TCL_TCLSH=C:/Tcl/bin/tclsh83.exe
- TCL_LIBRARY=C:/Tcl/lib/tcl83.lib
- TCL_INCLUDE_PATH=C:/Tcl/include
- LIBRARY_OUTPUT_PATH=C:/vtkbin/bin
- EXECUTABLE_OUTPUT_PATH=C:/vtkbin/bin
- CMAKE_USE_WIN32_THREADS=ON

- CMAKE_MAKE_PROGRAM=msdev
- CMAKE_EXTRA_LINK_FLAGS=/STACK:10000000
- CMAKE_CXX_FLAGS_RELWITHDEBINFO=/MD /Zi /O2
- CMAKE_CXX_FLAGS_RELEASE=/MD /O2
- CMAKE_CXX_FLAGS_MINSIZEREL=/MD /O1
- CMAKE_CXX_FLAGS_DEBUG=/MDd /Zi /Od /GZ
- CMAKE_CXX_COMPILER=cl
- CMAKETEST_COMMAND=NOTFOUND
- BUILD_DOCUMENTATION=OFF
- BUILD_EXAMPLES=ON
- BUILD_SHARED_LIBS=ON[1]
- BUILD_TESTING=ON
- CMAKE_CXX_FLAGS=/W3 /Zm1000 /GX /GR
- DART_ROOT=NOTFOUND
- OPENGL_LIBRARY=opengl32
- VTK_DATA_ROOT=C:/VTKData
- VTK_USE_ANSI_STDLIB=OFF[2]
- VTK_USE_HYBRID=ON
- VTK_USE_MATROX_IMAGING=OFF
- VTK_USE_PARALLEL=OFF
- VTK_USE_PATENTED=ON
- VTK_USE_RENDERING=ON
- VTK_USE_VIDEO_FOR_WINDOWS=OFF
- VTK_USE_VOLUMEPRO=OFF
- VTK_WRAP_JAVA=OFF
- VTK_WRAP_PYTHON=OFF
- VTK_WRAP_TCL=ON

Press `Configure` and resolve any fields marked with red, repeat until everything is ok. After this press `OK`, this will cause directory `C:\vtkbin` to be created and filled with a project ready to be built.

[1] This causes the VTK dlls to be built

[2] When compiling using the standard library I have experienced that visual studio fails with a heap error when building C++ projects.

5 Building VTK

Open the project workspace `C:\vtkbin\VTK.dsw` with Visual C++. Build the debug and the release version of the `ALL_BUILD` project.[3]

6 Setting Up the System

Add `C:/vtkbin/bin/release` to the system path. Add `TCLLIBPATH=c:/vtk/wrapping/tcl` as an environment variable (be carefull to use forward slashes. On windows 2000 this can be done using `start | settings | control panel | system | Advanced | Environment variables`. This can be verified by starting `wish` and issuing the command `puts $auto_path`. The above mentioned path shall be shown.

To test the installation double click on `C:/VTK/Examples/GUI/Tcl/MaceTk.tcl`.

If `*.tcl` files are not associated with a program then right click `C:/VTK/Examples/GUI/Tcl/MaceTk.tcl` and choose "open with"', select `C:/vtkbin/bin/Release/vtk.exe` as program and confirm that this program shall always be used with `.tcl` files. Alternatively use `c:/tcl/bin/wish83.exe` to open `.tcl` files.

7 Setting up Visual C++

The following shall only be done once, not for every new project.[4]

Start Visual C++ and open the `Tools | Options | Directories` menu. Add the following to the include directories:

- `C:\VTK\Common\`
- `C:\VTK\Filtering\`
- `C:\VTK\Graphics\`
- `C:\VTK\Hybrid\`
- `C:\VTK\Imaging\`
- `C:\VTK\IO\`
- `C:\VTK\Parallel\`
- `C:\VTK\Patented\`
- `C:\VTK\Rendering\`
- `C:\vtkbin\`

Note that `C:\vtkbin\` is added since `vtkConfigure.h` is placed there.

[3] It is in most cases only necessary to build the release version

[4] If more than one version of VTK are installed and they are used for different projects, this method is not recommended. Set up the include path for each project instead.

8 Starting a new project

There are several ways to start a new project using VTK. The method recommended by Kitware is to use CMake to start new projects. Look at the examples or in the books [1, 2] to see how this is done. In the following a manual non-CMake method is described.

Start Visual C++ and create a new project of type `Win32 Console Application` in the next window choose `a simple application` and finish the wizard.

Open `Project settings | C/C++ | Code Generation` and set the run-time library to `Debug Multithreaded DLL` for the debug configuration and `Multithreaded DLL` for the release configuration.

Open `Project settings | Link | Input` and add the following to the Object/library modules:

`vtkCommon.lib vtkFiltering.lib vtkGraphics.lib vtkHybrid.lib vtkImaging.lib vtkIO.lib vtkPatented.lib vtkRendering.lib glaux.lib opengl32.lib.`

this is done in both debug and the release configuration (All configurations).

Set the `Additional Library Path` to `C:\vtkbin\bin\release` in the release configuration and to `C:\vtkbin\bin\debug` in the debug configuration.[5]

In the current version of VTK there is no `vtk.h`, instead all needed header files must be included.

9 Acknowledgements

Thanks to the following people for comments:

- Allan Reinhold
- Klaus Baggesen Hilger
- Tim Hutton
- Mark Wrobel
- Will Schroeder

References

[1] W. J. Schroeder, L. S. Avila, K. M. Martin, W. A. Hoffman, and C. C. Law. *The Visualization Toolkit User's Guide*. Kitware, 2001.

[2] W.J. Schroeder, K. Martin, and W.E. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Prentice Hall, 1997.

[5] If only the release version of VTK is built, then let both paths point to the release directory.