無料電子ブック

学習

vtk

Free unaffiliated eBook created from
Stack Overflow contributors.

#vtk

1
1: vtk 2
2
Examples 2
2
Windows 7 2
2
2
2
3
4
5
5
2: 7
Examples 7
Hello World 7
7
9
10

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: vtk

It is an unofficial and free vtk ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official vtk.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: vtkをいめる

このセクションでは、vtkのとがなぜそれをいたいのかをします。

また、vtkのきなテーマについてもし、するトピックにリンクするがあります。 vtkのドキュメンテーションはしいものなので、それらのトピックのバージョンをするがあります。

Examples

インストールまたはセットアップ

Windows 7でのビルドとインストール

- のソースからVTKをビルドしたいは、 ここからgitがです。また、zipとしてコードのスナップショットをダウンロードし、ディスクドライブにすることもできます
- CMake
- Microsoft Visual Studio 2015
- たくさんのき - なにくにはなくともGBのがです

をする

- はをきれいにつのがきで、は3つのフォルダをします。

```
c:\vtk              #
c:\vtk\src          # 'code base' folder
c:\vtk\build        # 'out of source' build folder
c:\vtk\install      # 'install folder' where the 'installed' files will reside
```

- gitメソッドをしているは、
    - コマンドプロンプトをく
    - ディレクトリをする`cd c:\vtk\src`
    - gitリポジトリ`git clone https://gitlab.kitware.com/vtk/vtk.git`。 インターネットのによってはがかかることがあります
    - プロキシのでしているは、gitをしてするがあります。それをうについては、 このをしてください。
- zipメソッドをしているは、ソースコードを`c:\vtk\src`にします
- CMake GUIをする
- `Where is the source code: c:\vtk\src`をし`Where is the source code:`
- `Where to build the binaries: c:\vtk\build`をし`c:\vtk\build Where to build the binaries:`

- 「 `Configure`をクリックし、なジェネレーターとして`Visual Studio 2015`をします。
- いくつかのオプションがされます
- はにのビルドのためにのをします
  - `CMAKE_INSTALL_PREFIX` = `c:\vtk\install`
  - `BUILD_SHARED_LIBS`チェックされました
  - `BUILD_DOCUMENTATION` unticked
  - `BUILD_TESTING` unticked
  - `CMAKE_CXX_MP_FLAG`チェックされました。これにより、すべてのCPUコアマルチコア/マルチプロセッサシステムがされ、ビルドがスピードアップされます
- ヒットをするすべてのいエントリがくなるまでエラーをするように`Configure`
- ヒット`Generate`
- じるCMake GUI

- がしたは、
  - Visual Studioソリューション

    ```
    c:\vtk\build\vtk.sln
    ```

  - プロジェクトファイルの -

    ```
    ALL_BUILD.vcxproj
    INSTALL.vcxproj
    vtkCompileTools.vcxproj
    VTKData.vcxproj
    ZERO_CHECK.vcxproj
    ```

- これは、コマンドラインからでもIDEをってでもビルドできます
- はそれがにく、よりないRAMをするので、コマンドラインをむ
- コマンドラインの
  - `Developer Command Prompt For Visual Studio 2015`する
  - ディレクトリをする `cd c:\vtk\build`
  - msbuildをする
    - デバッグビルド
      - `msbuild /p:Configuration=Debug ALL_BUILD.vcxproj`
      - `msbuild /p:Configuration=Debug INSTALL.vcxproj`
    - リリースビルド
      - `msbuild /p:Configuration=Release ALL_BUILD.vcxproj`
      - `msbuild /p:Configuration=Release INSTALL.vcxproj`
- IDEの
  - Visual Studio 2015で`VTK.sln`をき、 `INSTALL.vcxproj`をビルドし`INSTALL.vcxproj`
  - このは、IDEがソリューションにリストされているプロジェクトのインテリセンスをしめるため、はくなります
- `c:\vtk\install`しいフォルダがされるはずです
  - `bin` にはdllファイルがまれています
  - `lib` にはlibファイルがまれています
  - `cmake`
  - `share`

    - `include` はヘッダファイルがまれています

ビルドの

- Visual C ++プロジェクトでVTKをするには、
    - コンパイラのヘッダーファイルのパスに`c:\vtk\include\vtk-<version>`をめるようにします。
    - `c:\vtk\lib`をむようにリンカー・ライブラリー・ファイルパスをします。
    - な`.lib`ファイルにリンクするようにリンカーをする
    - なDLLをフォルダにコピーする
- はすべての4つのタスク`c:\vtk\vtk.vsprops`をするためのさなファイルを`c:\vtk\vtk.vsprops`

```
<?xml version="1.0" encoding="UTF-8"?>
<Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
    <PropertyGroup>
        <VTK_ROOT_DIR>$(MSBuildThisFileDirectory)</VTK_ROOT_DIR>
        <VTK_BIN_DIR>$(VTK_ROOT_DIR)\bin</VTK_BIN_DIR>
        <VTK_INC_DIR>$(VTK_ROOT_DIR)\include\vtk-7.0</VTK_INC_DIR>
        <VTK_LIB_DIR>$(VTK_ROOT_DIR)\lib</VTK_LIB_DIR>
    </PropertyGroup>

    <PropertyGroup>
        <BuildDependsOn>CopyVTKBinariesList;$(BuildDependsOn);</BuildDependsOn>
    </PropertyGroup>

    <Target Name="CopyVTKBinariesList">
        <ItemGroup>
            <VtkBinaries Include="$(VTK_BIN_DIR)\*.dll" />
        </ItemGroup>
        <Copy SourceFiles="@(VtkBinaries)"
              DestinationFiles="@(VtkBinaries-
>'$(OutDir)\%(RecursiveDir)%(Filename)%(Extension)')"
              SkipUnchangedFiles="true" />
    </Target>

    <ItemDefinitionGroup>
      <ClCompile>

<AdditionalIncludeDirectories>$(VTK_INC_DIR);%(AdditionalIncludeDirectories)</AdditionalIncludeDirecto

      </ClCompile>
      <Link>

<AdditionalLibraryDirectories>$(VTK_LIB_DIR);%(AdditionalLibraryDirectories)</AdditionalLibraryDirecto

        <AdditionalDependencies>vtkalglib-7.0.lib;vtkChartsCore-7.0.lib;vtkCommonColor-
7.0.lib;vtkCommonComputationalGeometry-7.0.lib;vtkCommonCore-7.0.lib;vtkCommonDataModel-
7.0.lib;vtkCommonExecutionModel-7.0.lib;vtkCommonMath-7.0.lib;vtkCommonMisc-
7.0.lib;vtkCommonSystem-7.0.lib;vtkCommonTransforms-7.0.lib;vtkDICOMParser-
7.0.lib;vtkDomainsChemistry-7.0.lib;vtkDomainsChemistryOpenGL2-7.0.lib;vtkexoIIc-
7.0.lib;vtkexpat-7.0.lib;vtkFiltersAMR-7.0.lib;vtkFiltersCore-7.0.lib;vtkFiltersExtraction-
7.0.lib;vtkFiltersFlowPaths-7.0.lib;vtkFiltersGeneral-7.0.lib;vtkFiltersGeneric-
7.0.lib;vtkFiltersGeometry-7.0.lib;vtkFiltersHybrid-7.0.lib;vtkFiltersHyperTree-
7.0.lib;vtkFiltersImaging-7.0.lib;vtkFiltersModeling-7.0.lib;vtkFiltersParallel-
7.0.lib;vtkFiltersParallelImaging-7.0.lib;vtkFiltersProgrammable-7.0.lib;vtkFiltersSelection-
7.0.lib;vtkFiltersSMP-7.0.lib;vtkFiltersSources-7.0.lib;vtkFiltersStatistics-
```

```
7.0.lib;vtkFiltersTexture-7.0.lib;vtkFiltersVerdict-7.0.lib;vtkfreetype-7.0.lib;vtkGeovisCore-
7.0.lib;vtkglew-7.0.lib;vtkhdf5-7.0.lib;vtkhdf5_hl-7.0.lib;vtkImagingColor-
7.0.lib;vtkImagingCore-7.0.lib;vtkImagingFourier-7.0.lib;vtkImagingGeneral-
7.0.lib;vtkImagingHybrid-7.0.lib;vtkImagingMath-7.0.lib;vtkImagingMorphological-
7.0.lib;vtkImagingSources-7.0.lib;vtkImagingStatistics-7.0.lib;vtkImagingStencil-
7.0.lib;vtkInfovisCore-7.0.lib;vtkInfovisLayout-7.0.lib;vtkInteractionImage-
7.0.lib;vtkInteractionStyle-7.0.lib;vtkInteractionWidgets-7.0.lib;vtkIOAMR-7.0.lib;vtkIOCore-
7.0.lib;vtkIOEnSight-7.0.lib;vtkIOExodus-7.0.lib;vtkIOExport-7.0.lib;vtkIOGeometry-
7.0.lib;vtkIOImage-7.0.lib;vtkIOImport-7.0.lib;vtkIOInfovis-7.0.lib;vtkIOLegacy-
7.0.lib;vtkIOLSDyna-7.0.lib;vtkIOMINC-7.0.lib;vtkIOMovie-7.0.lib;vtkIONetCDF-
7.0.lib;vtkIOParallel-7.0.lib;vtkIOParallelXML-7.0.lib;vtkIOPLY-7.0.lib;vtkIOSQL-
7.0.lib;vtkIOVideo-7.0.lib;vtkIOXML-7.0.lib;vtkIOXMLParser-7.0.lib;vtkjpeg-7.0.lib;vtkjsoncpp-
7.0.lib;vtklibxml2-7.0.lib;vtkmetaio-7.0.lib;vtkNetCDF-7.0.lib;vtkNetCDF_cxx-
7.0.lib;vtkoggtheora-7.0.lib;vtkParallelCore-7.0.lib;vtkpng-7.0.lib;vtkproj4-
7.0.lib;vtkRenderingAnnotation-7.0.lib;vtkRenderingContext2D-
7.0.lib;vtkRenderingContextOpenGL2-7.0.lib;vtkRenderingCore-7.0.lib;vtkRenderingFreeType-
7.0.lib;vtkRenderingImage-7.0.lib;vtkRenderingLabel-7.0.lib;vtkRenderingLOD-
7.0.lib;vtkRenderingOpenGL2-7.0.lib;vtkRenderingVolume-7.0.lib;vtkRenderingVolumeOpenGL2-
7.0.lib;vtksqlite-7.0.lib;vtksys-7.0.lib;vtktiff-7.0.lib;vtkverdict-7.0.lib;vtkViewsContext2D-
7.0.lib;vtkViewsCore-7.0.lib;vtkViewsGeovis-7.0.lib;vtkViewsInfovis-7.0.lib;vtkzlib-
7.0.lib;%(AdditionalDependencies)</AdditionalDependencies>
      </Link>
    </ItemDefinitionGroup>
    <ItemGroup />

</Project>
```

- のvspropsファイルは、なすべてのdllを`c:\vtk\bin`フォルダにコピーします。
- DLLをつけることができるかどうかをするのは、デバッグセッションにPATHをし、VTKバイナリパスををみむときににするディレクトリにすることです。これをうには、のフラグメントを`CopyVTKBinariesList`タスクのわりにすることができます。

  ```
  <PropertyGroup>

  <LocalDebuggerEnvironment>PATH=$(VTK_BIN_DIR);%PATH%;$(LocalDebuggerEnvironment)</LocalDebuggerEn

  </PropertyGroup>
  ```

- なのために、 ウォーカーのようなツールをして、どのDLLとそれらのがされているかをし、のものだけをバンドルすることができます。
- Visual C ++プロジェクトでpropsファイルをするには、Visual Studioでプロパティマネージャツールメニュー=>プロパティマネージャをするか、vcxprojをテキストエディタでし、のをします。 `<Import Project="C:\vtk\vtk.vsprops" />`のプロジェクトインポートのに`<Import Project="C:\vtk\vtk.vsprops" />`します。

- ディスクスペースをしたいは、 `c:\vtk\build`フォルダをできますが、vtkにデバッグできないというがあります

アンインストール

- もうVTKにしたくないは、に`c:\vtk`フォルダをしてください

MacOSXとUnix

1. CMakeのバージョンをここにインストールしてください
2. ここでのVTKをダウンロードしてください 。
3. VTKのビルドディレクトリをする`mkdir <path_to_build_directory`
4. `ccmake <path_to_VTK_directory -G "UNIX Makefiles" \ -DVTK_USE_QVTK:BOOL=ON \ -DVTK_USE_CARBON:BOOL=ON \ -DCMAKE_INSTALL_PREFIX=/usr/local \ -DVTK_USE_GUISUPPORT:BOOL=ON` またはGUIをして`ccmake <path_to_VTK_directory`
5. ビルドディレクトリにし、`make -j`あなたがするはありません `-j`が、コンパイルはにいです 。
6. に`make installmake install`

オンラインでvtkをいめるをむ https://riptutorial.com/ja/vtk/topic/5182/vtkをいめる

2: こんにちは

Examples

Hello Worldの

```
#include <vtkAutoInit.h>

VTK_MODULE_INIT(vtkRenderingOpenGL2);
VTK_MODULE_INIT(vtkRenderingFreeType);
VTK_MODULE_INIT(vtkInteractionStyle);

#include <vtkSmartPointer.h>
#include <vtkTextActor.h>
#include <vtkRenderer.h>
#include <vtkRenderWindow.h>
#include <vtkRenderWindowInteractor.h>

int main(int /*argc*/, char ** /*argv*/)
{
    auto textActor = vtkSmartPointer<vtkTextActor>::New();
    textActor->SetInput("Hello World");

    auto renderer = vtkSmartPointer<vtkRenderer>::New();
    renderer->AddActor(textActor);
    renderer->ResetCamera();

    auto interactor = vtkSmartPointer<vtkRenderWindowInteractor>::New();

    auto renderWindow = vtkSmartPointer<vtkRenderWindow>::New();
    renderWindow->AddRenderer(renderer);
    renderWindow->SetInteractor(interactor);

    interactor->Start();

    return 0;
}
```

す

```
#include <vtkAutoInit.h>

VTK_MODULE_INIT(vtkRenderingOpenGL2);
VTK_MODULE_INIT(vtkRenderingFreeType);
VTK_MODULE_INIT(vtkInteractionStyle);
```

VTKデザインでは、 ファクトリメソッドデザインパターンをして、 `<ClassName>::New()` メソッドをしてvtkObjectクラスのしいインスタンスをします。これにより、にプラットフォームのをして、なインタフェースをたすことができます。

このみがするためには、ファクトリクラスをvtkインフラストラクチャでできるように、を「」す

るがあります。このトピックにするは、こちらをご ください 。

VTK_MODULE_INITは、なモジュール/ライブラリこのではvtkRenderingOpenGL2 、 vtkRenderingFreeType 、 vtkInteractionStyle をにするためにされるマクロです。モジュールのにすると、 <ClassName>::New()びしでNULLれ、ランタイムエラーがされNULL 。

```
#include <vtkSmartPointer.h>
#include <vtkTextActor.h>
#include <vtkRenderer.h>
#include <vtkRenderWindow.h>
#include <vtkRenderWindowInteractor.h>
```

vtkSmartPointerロールは、 vtkObjectクラスインスタンスのライフタイムをするカウントをするで 、 std::unique_ptrロールにています。

vtkTextActorは、にをするためにできるシンプルなクラスです。

vtkRendererは、シーンのコンテンツをするクラスです。には、

- vtkActor2Dからした2Dアクター
- vtkProp3Dからした3Dアクター
- ボリューム vtkVolume
- カメラ vtkCamera
- ライト vtkLight

vtkRenderWindowは、プラットフォームにしないインターフェイスをするクラスです。

- レンダラーのコレクションをします。
- ユーザーのをして、 vtkRenderWindowInteractorをさらにするためにvtkRenderWindowInteractor にする

vtkRenderWindowInteractorは、ユーザマウス/キーボード/タイミングイベントをするアクションにマッピングするクラスです。には、 vtkInteractorStyleをしてさまざまなマッピングをします。

```
auto textActor = vtkSmartPointer<vtkTextActor>::New();
textActor->SetInput("Hello World");
```

テキストアクターをし、するをする

```
auto renderer = vtkSmartPointer<vtkRenderer>::New();
renderer->AddActor(textActor);
renderer->ResetCamera();
```

- レンダラーをする
- テキストアクターをする
- カメラのをリセットして、アクタがにされるようにします。

```
auto interactor = vtkSmartPointer<vtkRenderWindowInteractor>::New();
```

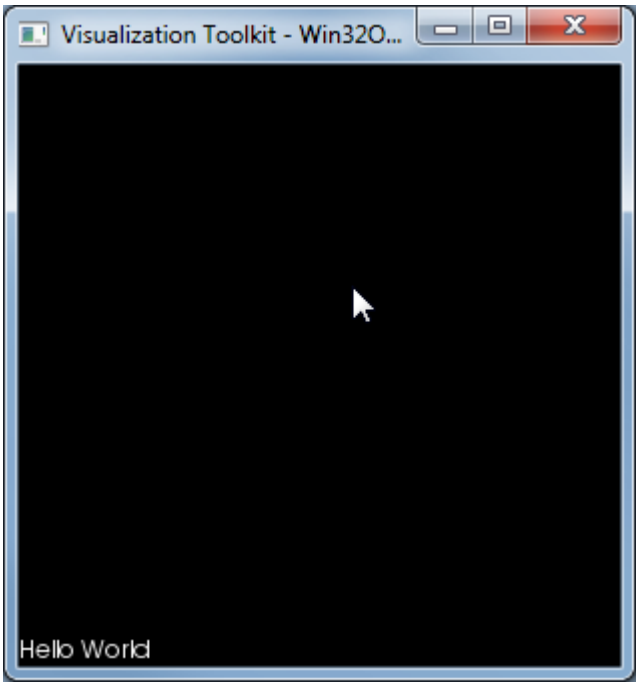
```
auto renderWindow = vtkSmartPointer<vtkRenderWindow>::New();
renderWindow->AddRenderer(renderer);
renderWindow->SetInteractor(interactor);
```

レンダリングするウィンドウをし、レンダラーをしてインタラクタをします。ファクトリは、/みのファクトリクラスにづいてなをにします

```
interactor->Start();
```

これは、ユーザーが qキーをするか、ウィンドウをじるときにのみされるブロッキングびしです。メッセージループをし、メッセージをディスパッチします。

これをすると、のようなウィンドウがされます。



ノート

このexeによってされたDLLのこのはのとおりです。

VTKCommonCore-7.0.DLL

VTKInteractionStyle-7.0.DLL

VTKRenderingCore-7.0.DLL

VTKRenderingFreeType-7.0.DLL

VTKRenderingOpenGL2-7.0.DLL

オンラインでこんにちはをむ https://riptutorial.com/ja/vtk/topic/5974/こんにちは

クレジット

S. No		Contributors
1	vtkをいめる	Community, LBes, Shreyas Murali
2	こんにちは	Shreyas Murali