Developer's Training Week

Download and Build Instructions

1. Overview

In preparation for the Developer's Training Week which will begin on June 9th, please follow the directions found in this document in order to download and build the necessary components based on the segments of the course you will attend. The table below indicates which components will need to be installed by which attendees. Once all components have been downloaded and installed, follow the build instructions that are provided per course topic.

	CMake	CVS	Tcl/Tk	Python	MPICH	Qt	VTK	ParaView	ITK
VTK Beginner	X	X	X				X		
VTK Advanced	X	X	X		X		X		
CMake	X								
ParaView	X			X	X	X		X	
ITK	X						X		X

2. Install CMake

All course attendees will require the latest version of CMake, which is 2.6.0. If you do not already have CMake 2.6.0 installed, you can obtain it from the CMake web site: http://www.cmake.org by clicking on the Download tab. Binaries are available for most

platforms, so download and install the binary appropriate for your particular operating system. If binaries are not available for your platform, download the CMake source code, and follow the instructions contained in the download. In addition, if binaries are not available for your platform, please let us know what operating system you will be using during the course.

3. Install CVS

Attendees for the VTK sessions will need to have CVS installed in order to obtain the source code to VTK 5.2. You can obtain CVS through http://www.nongnu.org/cvs/. Windows users may wish to use the CVS version available as part of Cygwin, which can be obtained here: http://sources.redhat.com/cygwin/, or WinCVS which has a nice graphics user interface which can be found here: http://www.wincvs.org/, or the Tortoise version of CVS that is integrated with Windows Explorer is available here: http://www.tortoisecvs.org/.

4. Obtain VTK

We will be using VTK 5.2 for the course, which has not yet been officially released. Therefore, you will need to obtain VTK 5.2 by anonymous CVS checkout with the correct branch tag. You can obtain read-only access to the CVS repository using the following commands.

```
cvs -d :pserver:anonymous@public.kitware.com:/cvsroot/VTK login
```

When prompted for a password, respond with vtk. Follow this command by checking out the source code:

```
cvs -d :pserver:anonymous@public.kitware.com:/cvsroot/VTK checkout –r
VTK-5-2 VTK
```

Once you execute this command you should have a VTK directory containing the source code for VTK 5.2. You will also need the VTK data for some of the exercises. You should get the matching VTK data for VTK 5.2 as shown below, again responding with a password of vtk.

```
cvs -d :pserver:anonymous@public.kitware.com:/cvsroot/VTKData login
cvs -d :pserver:anonymous@public.kitware.com:/cvsroot/VTKData checkout
–r VTK-5-2 VTKData
```

5. Install Tcl/Tk

We will be building VTK with Tcl-wrapping. If you do not already have Tcl installed on your computer, you can download ActiveTcl from the ActiveState web site: http://activestate.com/store/productdetail.aspx?prdGuid=f0cd6399-fefb-466e-ba17-220dcd6f4078.

6. Install Qt

To build ParaView with Qt without modifying ParaView, you'll need Qt version 4.2.3. If you'd like to instead build with the latest version of Qt, then you'll need to apply the modifications specified below. Please select one of the two options below for installing Qt.

Building ParaView With Qt 4.2.3

Building ParaView's user interface requires Trolltech's Qt, version 4.2.3. Qt is dual licensed. To compile ParaView, either the open or commercial version may be used. If you intend to make changes the ParaView's GUI and distribute the results, you must obtain a commercial license. The open source version of Qt is here ftp://ftp.trolltech.com/qt/source/ (remember, you need to 4.2.3 version for your platform). Note that if you are a Windows user, and you are not using the MinGW compiler, then you'll need to follow the instructions for getting our modified version of Qt here: http://paraview.org/Wiki/Obtaining_GPL%27ed_Qt_for_Windows. Commercial licenses can be purchased directly from TrollTech http://trolltech.com/. For more information on what can be done with the open source version, read this Wiki page: http://www.itk.org/Wiki/ParaView_III_and_Qt_licensing.

With Qt version 4.2.3 you will not need to make any modifications to ParaView after you obtain the source – it should build as is.

Building ParaView With Newer Versions of Qt

You may instead choose to build ParaView with the latest version of Qt (or any version greater than 4.2.x if you already have it). Note that Qt follows a dual licensing model and you can use either the commercial version or the GPL version to build ParaView, but if you want to make changes to ParaView's GUI and distribute the results you must obtain a commercial license. You can select from either the commercial or the open source version of Qt 4.4 here: http://trolltech.com/downloads. For more information on what can be done with the open source version and when you'd need a commercial license, please read this Wiki page: http://www.itk.org/Wiki/ParaView_III_and_Qt_licensing.

After you complete the instructions for obtaining the ParaView source code, you will need to make the following modifications in order to use this more recent version of Qt. Find the file ParaView3Src/Qt/CMakeLists.txt in the ParaView source tree and comment out lines 16 through 19 like this:

```
# STRING(REGEX MATCH "^4\\.2\\.[0-9]+" qt_version_tmp "${QTVERSION}")
# IF (NOT qt_version_tmp)
#  MESSAGE(SEND_ERROR "Qt ${QTVERSION} not supported.  Please use Qt
4.2.")
# ENDIF (NOT qt_version_tmp)
```

Once you make this modification, ParaView should build with the newer version of Qt.

7. Install MPI

Those attending the advanced VTK track or the ParaView session will need to install MPI. We recommend using MPICH2 which can be obtained here: http://www.mcs.anl.gov/research/projects/mpich2/.

8. Install Python

To enable scripting in ParaView, you will need Python. You can obtain it here: http://www.python.org/.

9. ParaView

Although ParaView is available as a binary installation, you should build ParaView from the source so that you can complete all the ParaView exercises in the course. To obtain the source, you can go here: http://paraview.org/New/download.html to get the ParaView 3.2.1 source in either tar or zip format. Alternatively, if you have CVS installed you can obtain ParaView using anonymous CVS checkout by executing:

```
cvs -d :pserver:anoncvs@www.paraview.org:/cvsroot/ParaView3 login
```

Respond with an empty password. Follow this command by checking out the source code:

```
cvs -d :pserver:anoncvs@www.paraview.org:/cvsroot/ParaView3 co –r
ParaView-3-2-1 ParaView3
```

Remember to make the necessary modifications indicated in the Qt section if you are going to be compiling with a version of Qt later than 4.2.x.

10. ITK

To build ITK, you will need to download the ITK 3.6 source code. You can get the Insight Toolkit source code for version 3.6 as either a zip or a tar file from: http://public.kitware.com/pub/itk/v3.6/ (get either the InsightToolkit-3.6.0.tar.gz file or the InsightToolkit-3.6.0.zip file). You can also download these files from SourceForge at: http://sourceforge.net/project/downloading.php?groupname=itk&filename=InsightToolkit-3.6.0.tar.gz&use_mirror=internap (for the tar file) or: http://sourceforge.net/project/downloading.php?groupname=itk&filename=InsightToolkit-3.4.0.zip&use_mirror=internap (for the zip file). Alternatively, if you have CVS you can get the source code via anonymous cvs access by executing the following:

```
cvs -d :pserver:anonymous@www.itk.org:/cvsroot/Insight login
```

Respond with the password insight. Then execute the following command to get the source:

```
cvs -d :pserver:anonymous@www.itk.org:/cvsroot/Insight co -r ITK-3-6
Insight
```

11. Build VTK (Beginner and Advanced)

Windows Configurations

Launch CMakeSetup.exe. Set "Where is the source code" to the top level of the VTK source tree. Set "Where to build the binaries" to your VTK binary location. Since you have not yet built VTK this directory does not yet exist (that is OK, CMake will create it for you – just specify a directory). Typically, if you source is located in C:\Course\VTK, then you would specify C:\Course\VTKBuild as your build directory. Do not use the same directory as both your source and build directory!

Once you have specified the source and build directories, click Configure. Select the compiler you will be using. You will need to set the following CMake variables as shown below.

- BUILD_SHARED_LIBS ON
- VTK_WRAP_TCL ON
- VTK_USE_PARALLEL ON

Depending on where you installed Tcl/Tk, you may need to specify the location of this library to VTK. If this is the case, CMake will let you know this during the configuration process. Once you have these variables set, click Configure again. When the OK button becomes active, click it to generate the appropriate project/workspace files for VTK.

Advanced VTK attendees: after configuring once, you will see a VTK_USE_MPI option appear that should be set to ON. CMake should find your MPI libraries that you installed – if not you may need to manually specify the paths.

Unix/Linux Configuration

Create a directory for your build. Do not try to build in the source directory. Typically if your source is located in /home/Course/VTK, you would create a directory /home/Course/VTKBuild as your build directory. Go into this build directory.

Run ccmake, passing it the path to the VTK source tree (i.e., ccmake /home/Course/VTK). Set the following CMake variables as shown below.

- BUILD_SHARED_LIBS ON

- VTK_WRAP_TCL ON
- VTK_USE_PARALLEL ON
- CMAKE_BUILD_TYPE Release

Depending on where you installed Tcl/Tk, you may need to specify the location of this library to VTK. If this is the case, CMake will let you know this during the configuration process. Once you have these variables set, press "C" to configure again. When the Generate (G) becomes an option in ccmake, select this option to generate the appropriate makefiles for VTK.

Compile VTK

When CMake finishes the configuration process and exits, compile VTK. On Windows, load the project/workspace file (in the VTK-bin directory) into your compiler environment, select a Release build, and compile VTK. On linux/unix type "make" inside the VTK-bin directory.

12. Build ParaView

The ParaView build instructions are available on the ParaView wiki which can be found here: http://paraview.org/Wiki/ParaView:Build_And_Install. Note that these instructions inform you that you can always get the binaries for ParaView – however, if you install only the ParaView binaries you will not be able to complete all of the ParaView exercises during the course since extending ParaView requires you to build ParaView from the source code.

13. Build ITK

Windows

Please follow these instructions to build ITK on Windows.

1. Run CmakeSetup.exe
2. Select the source directory where you extracted the source code of ITK
3. Select the directory directory where you want to build ITK
4. Click on the "Configure" button
5. Select your compiler (in case you have multiple compilers installed)
6. Set the BUILD_EXAMPLES variable to OFF

7. Set the BUILD_SHARED_LIBS variable to OFF
8. Set the BUILD_TESTING variable to OFF
9. Click on “Configure” again
10. Click on “OK”
11. Open the ITK.sln file in Visual Studio
12. Select the ALL_BUILD project
13. Select the Release compilation mode in the Build menu
14. Start the build. It should take about 20 minutes

GNU/Linux

Please follow these directions to build ITK on a GNU/Linux platform:

1. create a directory for hosting the binary build (e.g. mkdir ITKBin)
2. cd into that directory (e.g. cd ITKBin)
3. set in the “CC” environment variable the path to the C compiler you want to use.
4. set in the “CXX” environment variable the path to the C++ compiler you want to use.
5. invoke “ccmake” (note the double “c”) by passing as command line argument the name of the directory where you expanded the source code of ITK.
   (e.g. ccmake /home/ibanez/course/Insight/)
6. Set the BUILD_EXAMPLES variable to OFF
7. Set the BUILD_SHARED_LIBS variable to OFF
8. Set the BUILD_TESTING variable to OFF
9. Click on “c” for “Configure”
10. Click on “g” for “Generate”
11. When ccmake quits, you can start the build by typing “make”.
    The build should take about 20 minutes.

Using Both VTK and ITK

ITK course attendees should also have build VTK according to the instructions provided above. Please note that you should build both VTK and ITK in the same configuration (release or debug). We recommend building debug for the course since this will provide you with the best debugging opportunities during the exercises.

Since you are building shared libraries for VTK, you will have to add to your path the directory where these libraries are located. In MS-Windows you do this by following the instructions given in the ITK FAQ entry:

http://www.itk.org/Wiki/ITK_FAQ#How_to_set_Windows_DLL.27s_on_the_Path

Under **GNU/Linux** you should add to your environment variable "LD_LIBRARY_PATH ", the directory where your **VTK** shared libraries "*.so" files are located.