



Getting Started with ITK

(Instructions for Current Release)

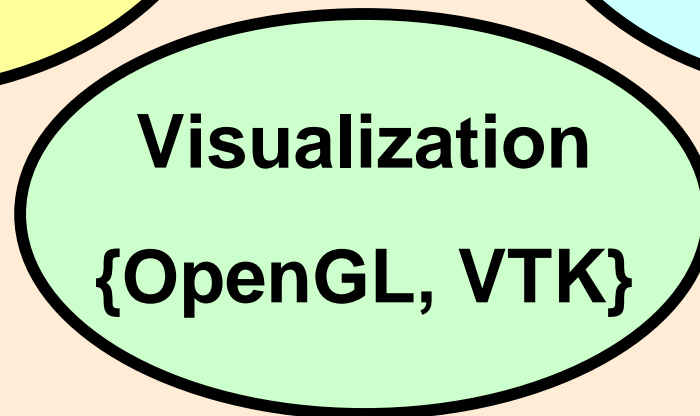
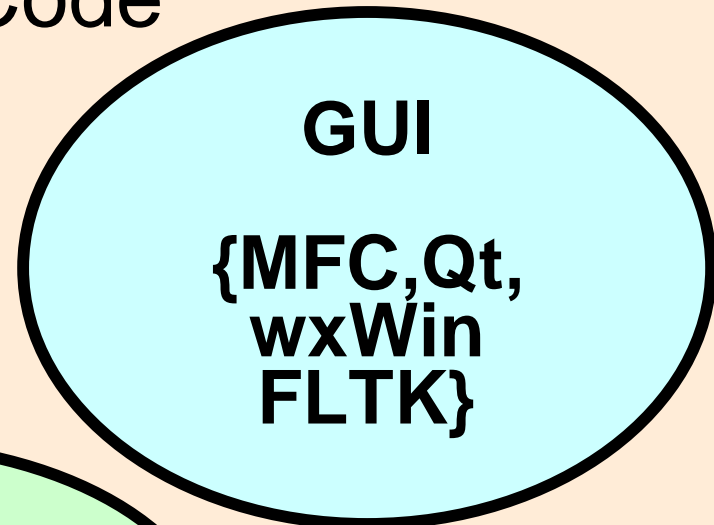
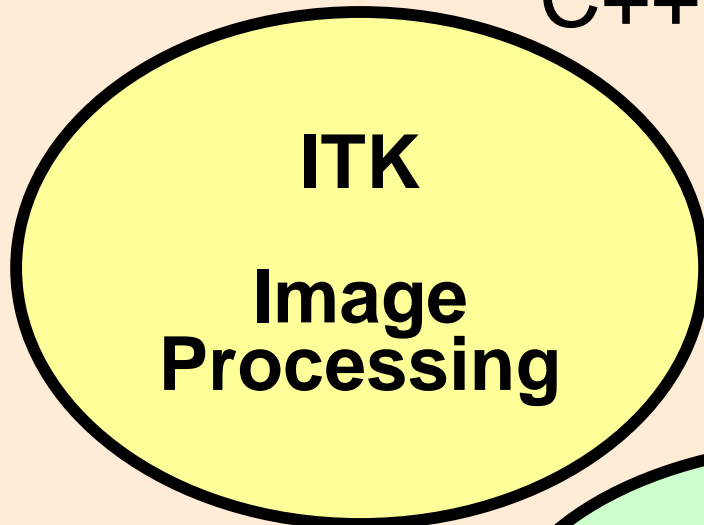
Luis Ibáñez
Will Schroeder
Brad King
Insight Software Consortium

What is ITK

- Image Processing
- Segmentation
- Registration
- No Graphical User Interface (GUI)
- No Visualization

How to Integrate ITK in you application

C++ Glue Code



What do I need ?

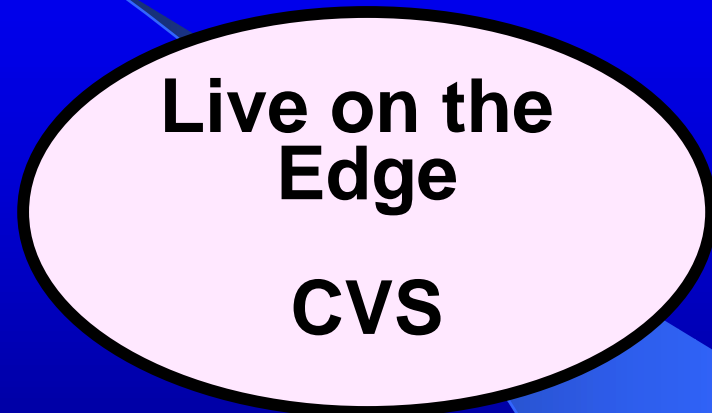
C++ Compiler

gcc 2.95 – 3.1
Visual C++ 6.0
Visual .NET
Intel 5.0
IRIX CC
Borland 5.0
Mac - gcc

CMake

www.cmake.org

Step 1. Download ITK



<http://www.itk.org>

A yellow rounded rectangle with a black border containing the URL "http://www.itk.org". Two yellow arrows point from the bottom of this rectangle to the "Insight.tgz" and "CVS anonymous" boxes.

Insight.tgz

A light blue rounded rectangle with a black border containing the text "Insight.tgz".

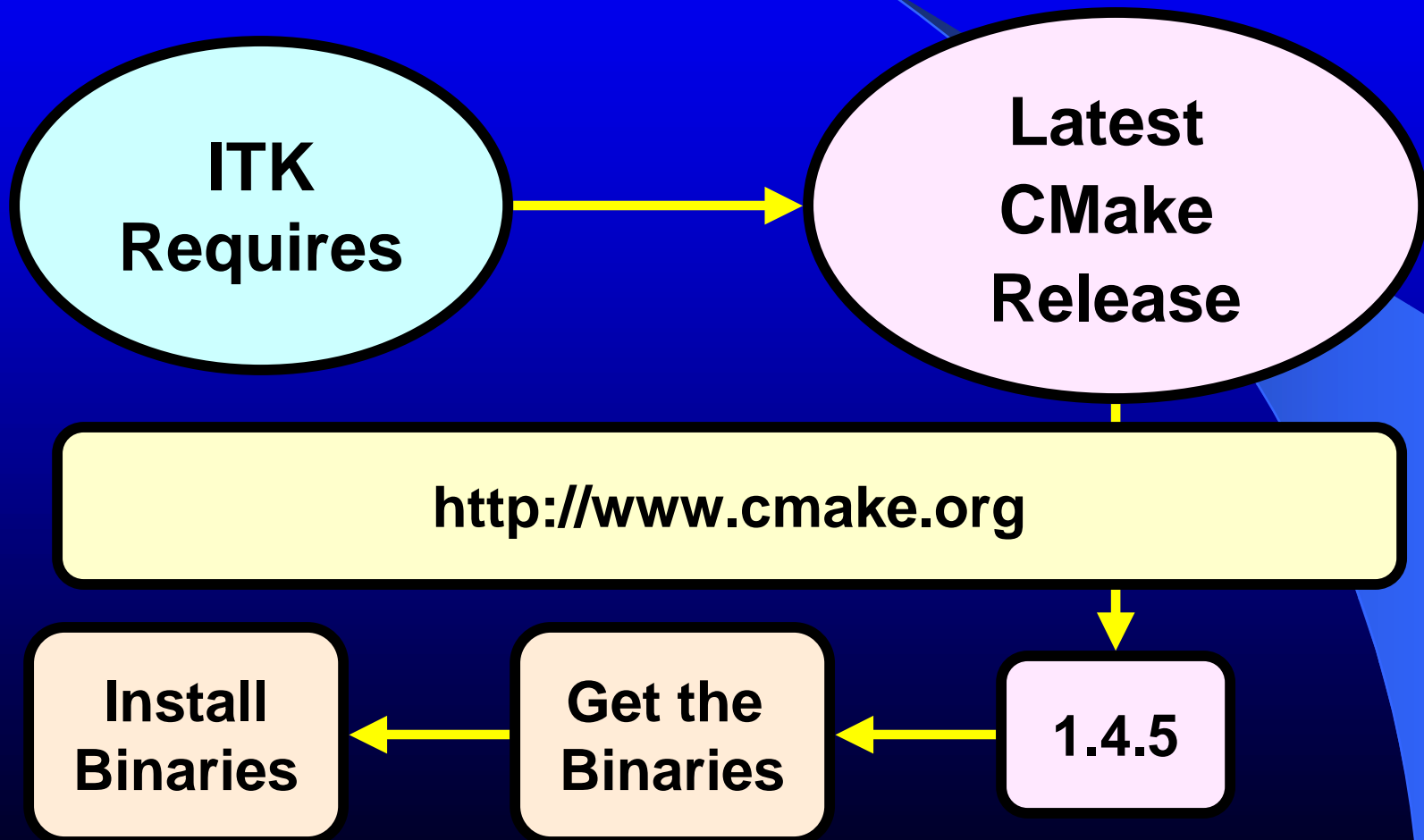
CVS anonymous

A light pink rounded rectangle with a black border containing the text "CVS anonymous".

Step 1: Download ITK

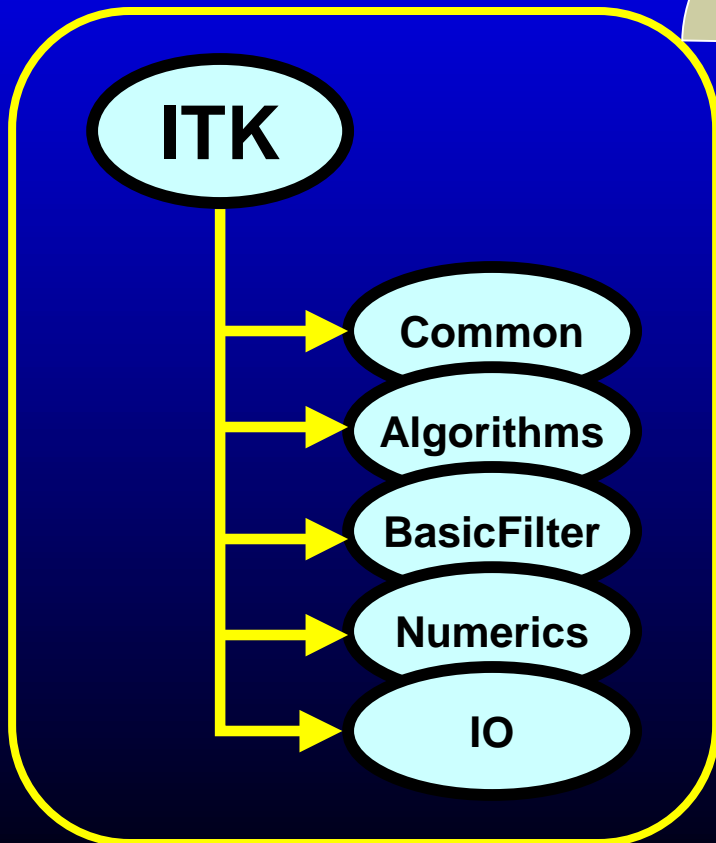
- Get packaged releases:
 - <http://www.itk.org/HTML/Download.php>
- Download from CVS (development version)
 - `cvs -d :pserver:anonymous@www.itk.org:/cvsroot/Insight login`
 - (respond with password "insight")
 - `cvs -d :pserver:anonymous@www.itk.org:/cvsroot/Insight co Insight`

Step 2. Download CMake



Step 3. Configure ITK

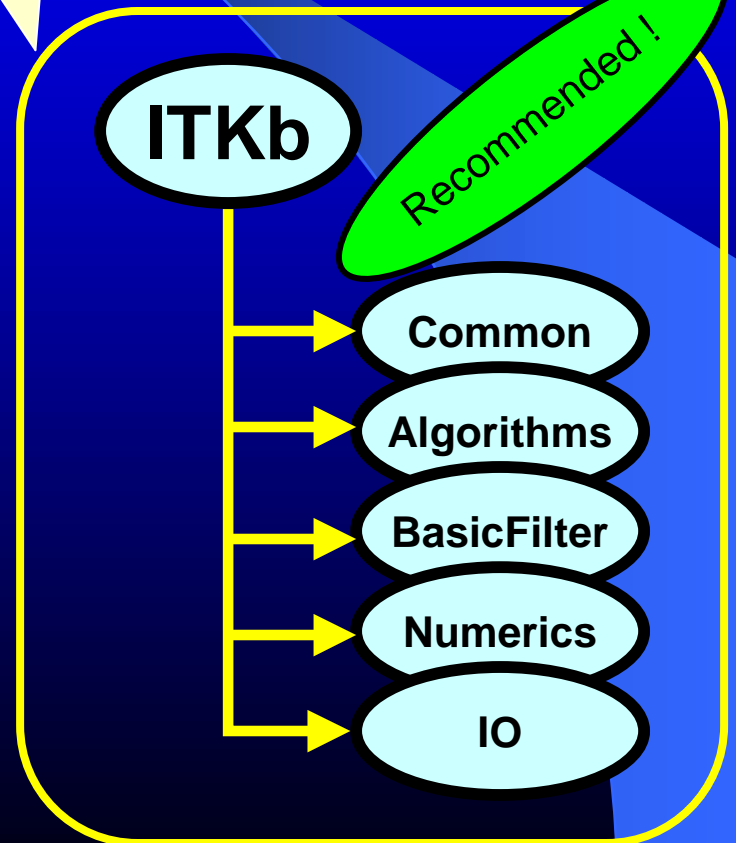
Source Tree



Out
Source Build

In
Source
Build

Binary Tree

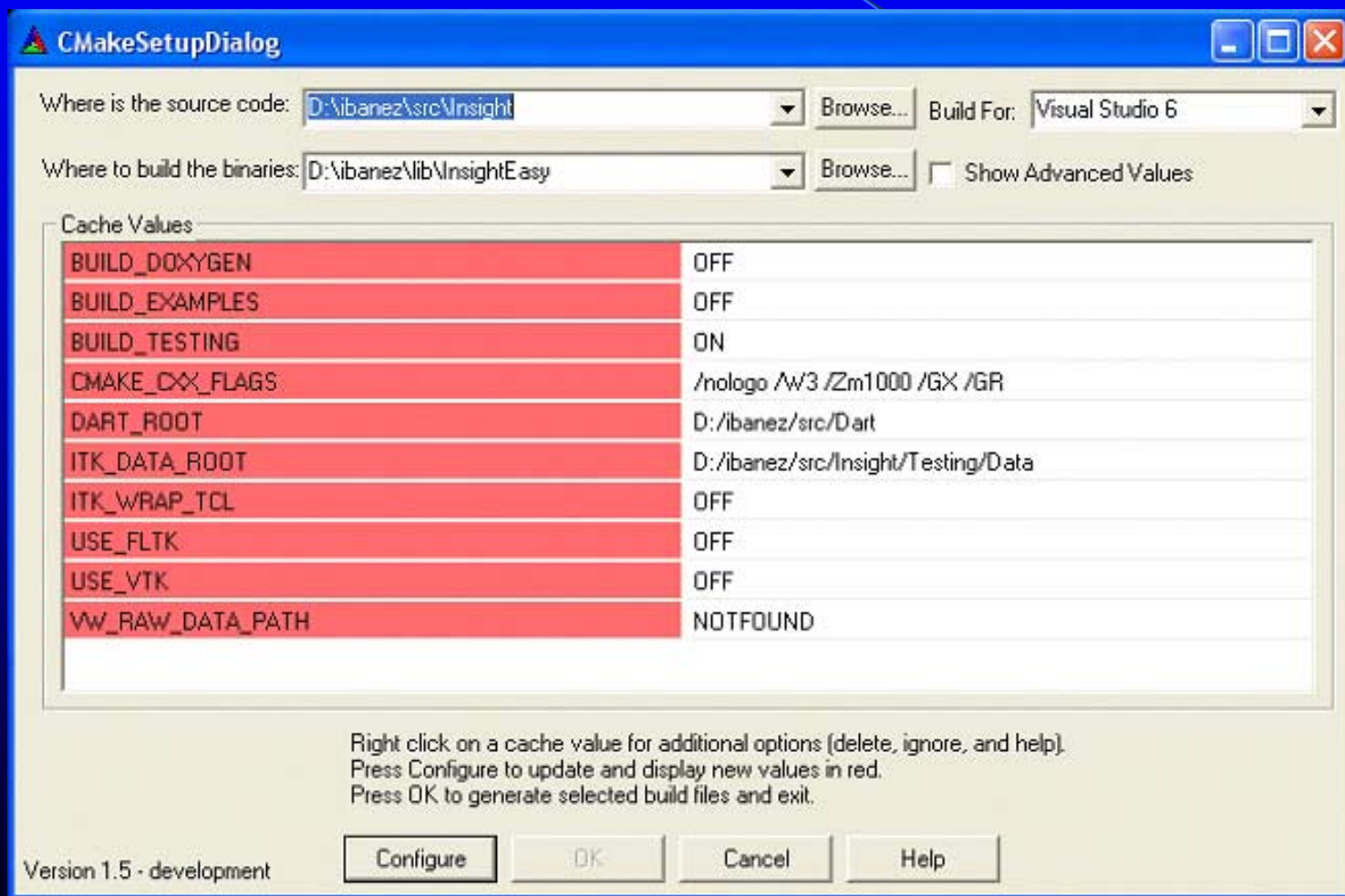


Step 3. Configure - Easy Start

- Run CMake
- Select the SOURCE directory
- Select the BINARY directory
- Select the COMPILER
- Select CONFIGURE and then OK buttons

(Note: CMake works iteratively. As new options are enabled, new CMake variables show up in red. Continue to select CONFIGURE until no red appears (no changes) then finally select OK to produce workspaces / makefiles)

Step 3. Configure - Easy Start



Step 3. Configure - Easy Start

- Disable BUILD_DOXYGEN
- Disable BUILD_EXAMPLES
- Enable BUILD_TESTING
- Disable USE_FLTK
- Disable USE_VTK
- Disable ITK_WRAP_TCL

(Note: BUILD_TESTING can be disabled; this will speed the build process but produce no executables (as referred to in Step 5 in the next slides))

Step 3. Configure - Easy Start

- Ignore CMAKE_CXX_FLAGS
- Ignore DART_ROOT
- Ignore ITK_DATA_ROOT
- Ignore VW_RAW_DATA_PATH

Step 4. Build Project

- Open ITK.dsw in the BINARY Directory (assuming MSVC compiler)
- Select ALL_BUILD project
 - Select configuration
 - Debug (*recommended initially*)
 - Release
 - RelWithDebugInfo
 - MinSizeRel
- Build...it will take about 1 hour ...
 - (*if BUILD_TESTING is disabled, less than 15 minutes*)

Step 4. Build Project

- Most of ITK classes are C++ Templates
- Basic libraries are small—they only contain non-templated classes
- Basic libraries are built in about 15 min

Step 5. Verify the Build

- Libraries and test Executables will be found in

ITK_BINARY / bin / { Debug, Release }

- *The actual location depends on the configuration chosen in MSVC compiler*
- *(Executables will be present only if BUILD_TESTING was enabled in CMake)*

Step 5. Verify the Build

The following libraries should be found

- ITKCommon
- ITKBasicFilters
- ITKAlgorithms
- ITKNumerics
- ITKFEM
- ITKIO
- ITKStatistics
- VXLNumerics
- itkpng
- itkzlib
- ITKMetaIO

Step 5. Verify the Build

The following executables should be found

- itkCommonTests
- itkBasicFiltersTests
- itkAlgorithmsTests
- itkNumericsTests
- itkIOTests

Step 5. Verify the Build

The following executables should be found

- itkSpatialObjectTests
- itkFEMTests
- itkStatisticsTests
- vnlTests

Step 5. Verify the Build

The following executables should be found

- itkCommonHeaderTest
- itkBasicFiltersHeaderTest
- itkAlgorithmsHeaderTest
- itkNumericsHeaderTest
- itkIOHeaderTest
- itkSpatialObjectHeaderTest

Step 5. Verify the Build

- Run ONE of the tests
- Murphy's Law guarantees that if there is only ONE test failing, it will be the one you randomly select !

Step 5. Verify the Build

- *Assuming that BUILD_TESTING was enabled:*
 - The test organization reflects the source tree structure
 - For example, in order to test the GradientImageFilter execute
`itkBasicFiltersTests.exe itkGradientImageFilter`
 - Preferred way to execute tests:
 - `cd` into BINARY directory (as set in CMake)
 - `ctest -R itkGradientImageFilter`
(-R says any test matching this string is executed;
without -R all tests are executed)
 - `ctest` is a companion program to CMake

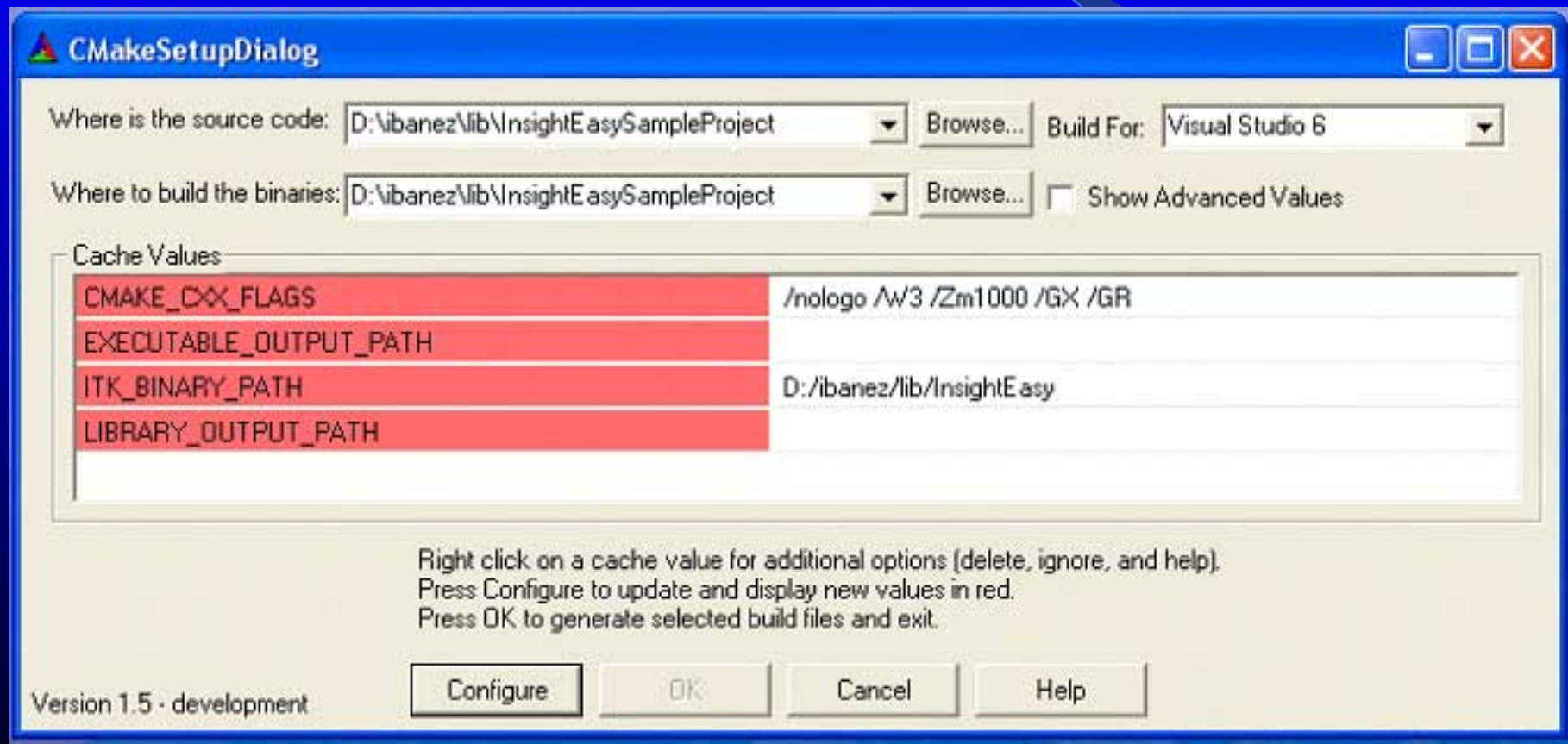
Step 6. Use ITK from an external Project

Copy
“SampleProject”
from the Examples
Directory
into another
directory

Run
CMake

- Select Source Dir
- Select Binary Dir
- Select Compiler

Step 6. Use ITK from an external Project



Step 6. Use ITK from an external Project

- Ignore CMAKE_CXX_FLAGS
- Ignore EXECUTABLE_OUTPUT_PATH
- Ignore LIBRARY_OUTPUT_PATH
- Point `ITK_BINARY_PATH` to the binary directory where ITK was built

Step 7. Build Sample Project

- Open SampleProject.dsw generated by CMake
- Select ALL_BUILD project
- Build it
...It will take about 10 seconds ...

Step 8. Run the example

- Locate the file itkSampleProject.exe
- Run it...
- It should produce the message:
Test Passed !

Step 9. Start your own project

- Create a clean new directory
- Write a CMakeLists.txt file
- Write a simple .cxx file
- Configure with CMake
- Build
- Run

Step 10. Writing CMakeLists.txt

```
PROJECT( myProject )
```

```
INCLUDE (${CMAKE_ROOT}/Modules/FindITK.cmake)
```

```
IF ( USE_ITK_FILE )
```

```
    INCLUDE(${USE_ITK_FILE})
```

```
ENDIF( USE_ITK_FILE )
```

```
ADD_EXECUTABLE( myProject myProject.cxx )
```

```
TARGET_LINK_LIBRARIES ( myProject ${ITK_LIBRARIES} )
```

Step 11. Writing myProject.cxx

```
#include "itkImage.h"
#include "itkImageFileReader.h"
#include "itkGradientMagnitudeImageFilter.h"

int main( int argc, char **argv ) {
    typedef itk::Image<unsigned short,2>           ImageType;
    typedef itk::ImageFileReader<ImageType>        ReaderType;
    typedef itk::GradientMagnitudeImageFilter<
                                                ImageType,ImageType> FilterType;

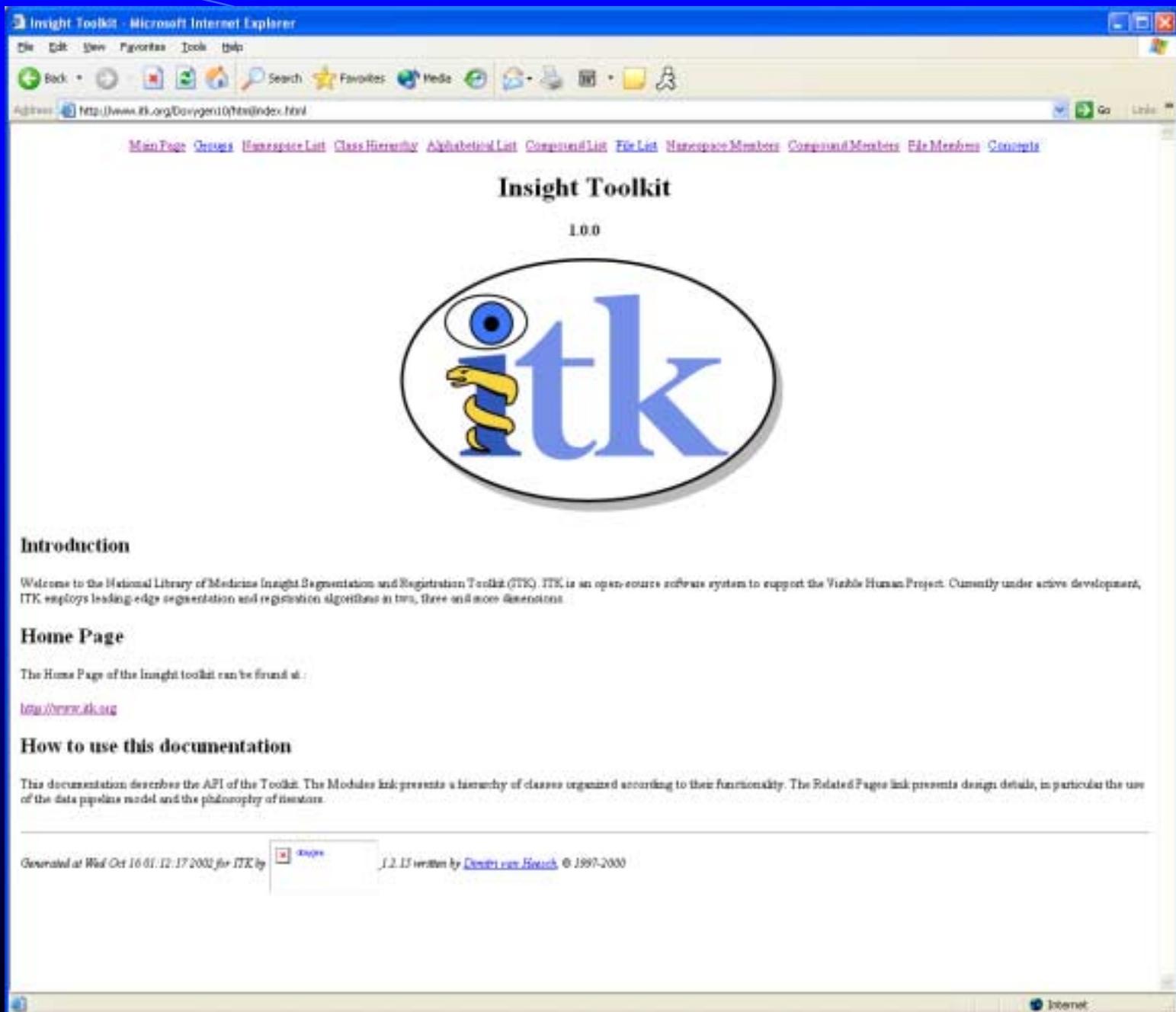
    ReaderType::Pointer reader = ReaderType::New();
    FilterType::Pointer filter = FilterType::New();

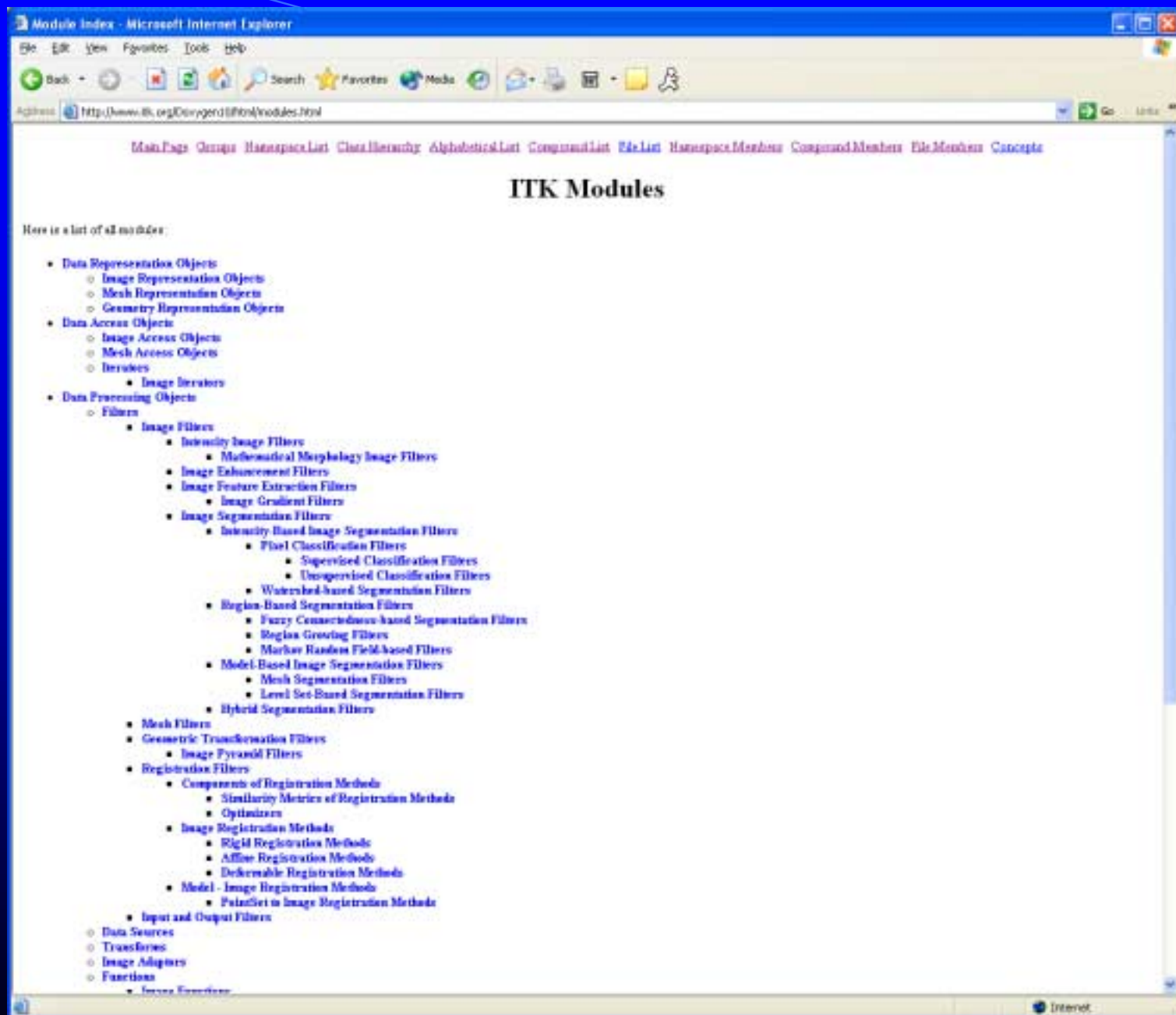
    reader->SetFileName( argv[1] );
    filter->SetInput( reader->GetOutput() );
    filter->Update();
    return 0;
}
```

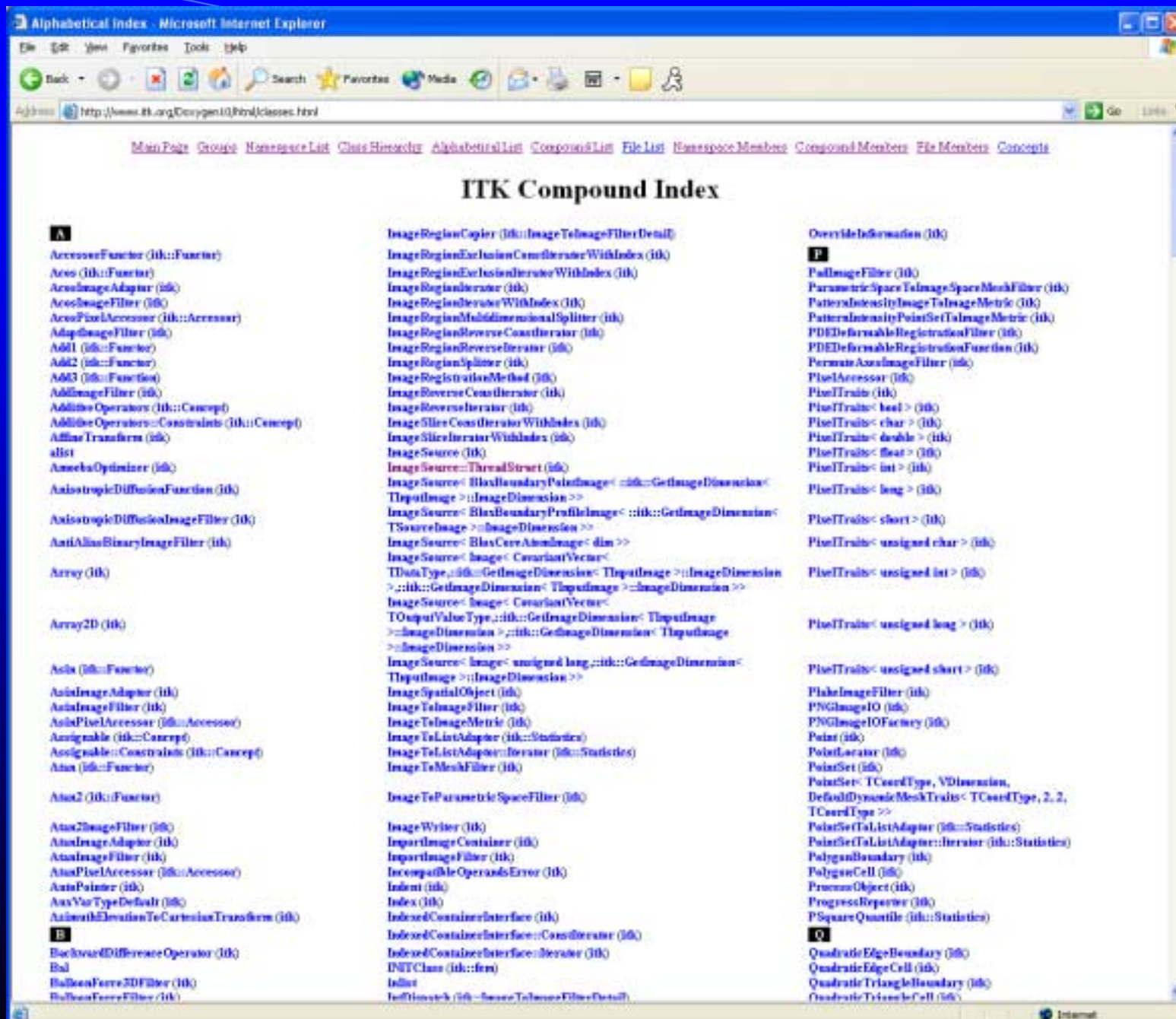
Step 12. How to find what you need

<http://www.itk.org/Doxygen10/html/index.html>

- Follow the link [Alphabetical List](#)
- Follow the link [Groups](#)
- Post to the [insight-users](#) mailing list







Additional Resources

- User Mailing List
 - <http://www.itk.org/mailman/listinfo/insight-users>
- Further Documentation
 - InsightDocuments (cvs checkout)

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

with password "insight"...then get the source/documentation as follows:

`cvs -d :pserver:anonymous@www.itk.org:/cvsroot/Insight checkout \
InsightDocuments`

The background is a solid blue gradient. A thin, light blue curved line starts from the top left and arcs towards the right. A larger, light blue wedge-shaped area is located on the right side, pointing towards the center.

Enjoy ITK !