What is ITK

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

- National Library of Medicine
- The Visible Human Project
- GE Corporate Research & Development
- Columbia University
- Kitware
- Insightful
- Brown University
- University of Pittsburgh
- University of Utah
- Penn VASTLab
- Penn
- GRASP Lab
ITK Developers

GE CRD
Bill Lorensen

Insightful
Lydia Ng

U Penn
Dimitris Metaxas

Harvard BWH*
Ron Kikinis

U Penn*
Jim Gee

Columbia U.*
Celina Imielinska

Kitware
Will Schroeder

UNC-CH
Stephen Aylward

U Tennessee
Ross Whitaker

U Pittsburgh*
George Stetten

U Utah*
Ross Whitaker

* indicates a subcontractor.
ITK by the Numbers

- Public Investment
  - $13 Million
- March 2000
  - First code check-in
- 2,040
  - # of nightly builds
- 1,289
  - tests run nightly
- 42
  - # of platforms (software + hardware)
- 1,647
  - # of C++ classes
- 2,314
  - # of files with code (Insight / Code directory)

Numbers updated on September 29th 2007
ITK by the Numbers

- 136K
  - # of lines of code (Insight / Code directory)
- 102K
  - # of lines of test code (Insight / Testing directory)
- 81.6%
  - Code coverage (gcov)
- 27K
  - # of lines of examples (Insight / Examples directory)
- 114K
  - # of lines of Applications (InsightApplications checkout)
- 328
  - weekly t-cons
- 82
  - unique developers

Numbers updated on September 29th 2007
ITK by the Numbers

- 1,380
  - # of users subscribed to the mailing-list
- 345
  - # of monthly emails in users-list (Oct’06 - Sep’07 average)
- 836
  - # of pages in the Software Guide PDF document
- 15,040
  - # of Software Guide PDF downloads (sourceforge)
- 3,393
  - # of monthly hits to the URL of the ITK Software Guide
- 4,169
  - # of Insight Applications downloads ITK 3.4 (sourceforge)
- 8,659
  - # of source code downloads ITK 3.4 (sourceforge)

Numbers updated on September 29th 2007
How to Integrate ITK in your application

C++ Glue Code

- ITK
  - Image Processing
- GUI
  - {MFC, Qt, wxWin, FLTK}
- Visualization
  - {OpenGL, VTK}
What do I need?

C++ Compiler
- GCC 2.95 – 4.1
- Visual C++ 6.0
- Visual C++ 7.0
- Visual C++ 7.1
- Visual C++ 8.0
- Intel 7.1
- Intel 8.0-9.0
- IRIX CC
- Borland 5.5
- Mac - gcc

CMake
www.cmake.org
Step 1. Download ITK

- Stability
  - Release tar files
- Live on the Edge
  - CVS

- http://www.itk.org
- Insight.tgz
- CVS anonymous
Copying ITK from the CD

/Source/
InsightToolkit-3.4.0.zip
InsightToolkit-3.4.0.tar.gz
Step 2. Download CMake

ITK Requires

Latest CMake Release

http://www.cmake.org

Install Binaries

Get the Binaries

2.4.7
Installing CMake from the CD

/CMake/
cmake-2.4.7-win32-x86.zip
cmake-2.4.7-win32-x86.exe
cmake-2.4.7-Linux-i386.tar.gz
Step 3. Configure ITK

Source Tree
- ITK
  - Common
  - Algorithms
  - BasicFilter
  - Numerics
  - IO

Binary Tree
- ITKb
  - Common
  - Algorithms
  - BasicFilter
  - Numerics
  - IO

Recommended!
Configuring ITK – MS-Windows

- Run CMake
- Select the **SOURCE** directory
- Select the **BINARY** directory
Configuring ITK – MS-Windows
Configuring ITK – MS-Windows

- Set BUILD_EXAMPLES to OFF
- Set BUILD_SHARED_LIBS to OFF
- Set BUILD_TESTING to OFF

- Click “Configure” to configure
- Select the Compiler that you want to use
- Click “OK” to generate project files
Configuring ITK – GNU/Linux

- Create the BINARY directory (mkdir)
- Change directory to the BINARY directory (cd)
- Set the environment variables CC and CXX
  - setenv CC /usr/bin/gcc; setenv CXX /usr/bin/g++ OR
  - export CC=/usr/bin/gcc; export CXX=/usr/bin/g++
- Type `ccmake` with argument the SOURCE directory
Configuring ITK – GNU/Linux
Configuring ITK – GNU/Linux

- Disable BUILD_EXAMPLES
- Disable BUILD_SHARED_LIBS
- Disable BUILD_TESTING

- Type “c” to configure
- Type “g” to generate the Makefiles
- Type “make” to start building
Building ITK
Building ITK

- Open ITK.dsw in the Binary Directory
- Select ALL_BUILD project
- Build it

...It will take about 15 minutes ...
Building ITK
Building ITK

- Open ITK.sln in the Binary Directory
- Select ALL_BUILD project
- Build it

...It will take about 15 minutes ...
Building ITK

- 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 Built

Libraries will be found in

ITK_BINARY / bin / { Debug, Release }
Step 5. Verify the Build

The following libraries should be there

- ITKCommon
- ITKBasicFilters
- ITKAlgorithms
- ITKNumerics
- ITKFEM

- ITKIO
- ITKStatistics
- ITKMetaIO
- itkpng
- itkzlib
Step 6. Use ITK from an external Project

Copy
“HelloWorld.cxx”
“CMakeLists.txt”
from the Examples/Installation Directory into another directory

Run CMake

- Select Source Dir
- Select Binary Dir
Using ITK – Hello World
Using ITK – Hello World

CMAKE_BACKWARDS_COMPATIBILITY
CMAKE_BUILD_TYPE
CMAKE_INSTALL_PREFIX
EXECUTABLE_OUTPUT_PATH
ITK_DIR
LIBRARY_OUTPUT_PATH

CMAKE_BACKWARDS_COMPATIBILITY: For backwards compatibility, what version of CMake commands and syntax
Press [enter] to edit option CMake Version 2.4 - patch 3
Press [c] to configure Press [g] to generate and exit
Press [h] for help Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
Step 6. Use ITK from an external Project

- accept the default in CMAKE_BACKBARD_COMPATIBILITY
- leave empty EXECUTABLE_OUTPUT_PATH
- leave empty LIBRARY_OUTPUT_PATH
- Set ITK_DIR to the binary directory where ITK was built
Step 7. Build Sample Project

- Open `HelloWorld.dsw` (or `.sln`) generated by CMake
- Select `ALL_BUILD` project
- Build it
  
  ...It will take about 3 seconds ...
Step 7. Build Sample Project
Step 8. Run the example

- Locate the file HelloWorld.exe
- Run it...
- It should produce the message:
  ITK Hello World!
Starting your own project

- Create a clean new directory
- Write a CMakeLists.txt file
- Write a simple .cxx file
- Configure with CMake
- Build
- Run
Step 9. Writing `CMakeLists.txt`

```cmake
PROJECT( myProject )

FIND_PACKAGE ( ITK REQUIRED )
IF ( ITK_FOUND )
    INCLUDE( ${ITK_USE_FILE} )
ENDIF( ITK_FOUND )

ADD_EXECUTABLE( myProject myProject.cxx )

TARGET_LINK_LIBRARIES ( myProject ITKCommon ITKIO)
```
Step 10. Writing myProject.cxx

```c++
#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 11. Run CMake
Step 12. How to find what you need

http://www.itk.org/ItkSoftwareGuide.pdf


- Follow the link Alphabetical List
- Follow the link Groups
- Post to the insight-users mailing list
Introduction

Welcome to the National Library of Medicine Insight Segmentation and Registration Toolkit (ITK). ITK is an open-source software system to support the Visible Human Project. Currently under active development, ITK employs leading-edge segmentation and registration algorithms in two, three and more dimensions.
ITK Modules

Here is a list of all modules:

- **Data Representation Objects**
  - Image Representation Objects
  - Mesh Representation Objects
  - Path Representation Objects
  - Geometry Representation Objects
- **Data Access Objects**
  - Image Access Objects
  - Mesh Access Objects
  - Iterators
    - Image Iterators
- **Objects Related to Tensor Images**
- **Data Processing Objects**
  - Filters
    - Image Filters
      - Intensity Image Filters
      - Mathematical Morphology Image Filters
      - Image Enhancement Filters
      - Image Feature Extraction Filters
ITK Class Index

A
AbortCheckEvent ( itk )
AbortEvent ( itk )
Abs ( itk::Function )
AbsImageAdaptor ( itk )
AbsImageFilter ( itk )
AbsoluteDifference2 ( itk::Functor )
AbsoluteDifferenceImageFilter ( itk )
AbsPerPixelAccessor ( itk::Accessor )
AccessorFunctor ( itk::Functor )
AccumulateImageFilter ( itk )
Acos ( itk::Functor )
AcosImageAdaptor ( itk )
AcosImageFilter ( itk )
AcosPerPixelAccessor ( itk::Accessor )
Alinear::Model::Calculate ( itk )

GrayscaleGeodesicDilateImageFilter ( itk )
GrayscaleGeodesicErodeImageFilter ( itk )
GrayscaleGrindPeakImageFilter ( itk )
GrayscaleMorphologicalClosingImageFilter
GrayscaleMorphologicalOpeningImageFilter
GreaterThanComparable ( itk::Concept )
GreaterThanComparable::Constraints ( itk )
GreenPixelAccessor ( itk )
GreyLevelCooccurrenceMatrixTextureCo
( itk::Statistics )
GroupSpatialObject ( itk )

HammingWindowFunction ( itk::Function )
HardLimitTransferFunction ( itk::Statistics )
hash ( itk )
hash< char * > ( itk )
HashTraits< int > ( itk )
The ITK Software Guide is freely available as a PDF document at

[www.itk.org/ItkSoftwareGuide.pdf](http://www.itk.org/ItkSoftwareGuide.pdf)

Its paper version can be ordered from Amazon.com and from Kitware’s e-store.
Enjoy ITK!