NA-MIC
National Alliance for Medical Image Computing
http://www.na-mic.org

Hello CLI:
contributing an algorithm into Slicer 4

Nadya Shusharina, Greg Sharp

Massachusetts General Hospital

nshusharina@partners.org

NA-MIC Tutorial Contest: Summer 2013

Learning Objective

This step by step tutorial leads you through developing command line interface (CLI) for Slicer 4 (*http://www.slicer.org*)

- Getting ready
- Building a template module
- Building module for image thresholding

National Alliance for Medical Image Computing
http://www.na-mic.org

Pre-requisite

- Slicer is an open-source software for segmentation, registration and visualization of medical imaging data

- The platform is developed through a multi-institution effort of several NIH funded large-scale consortia

- Slicer is for medical research only, and is not FDA approved

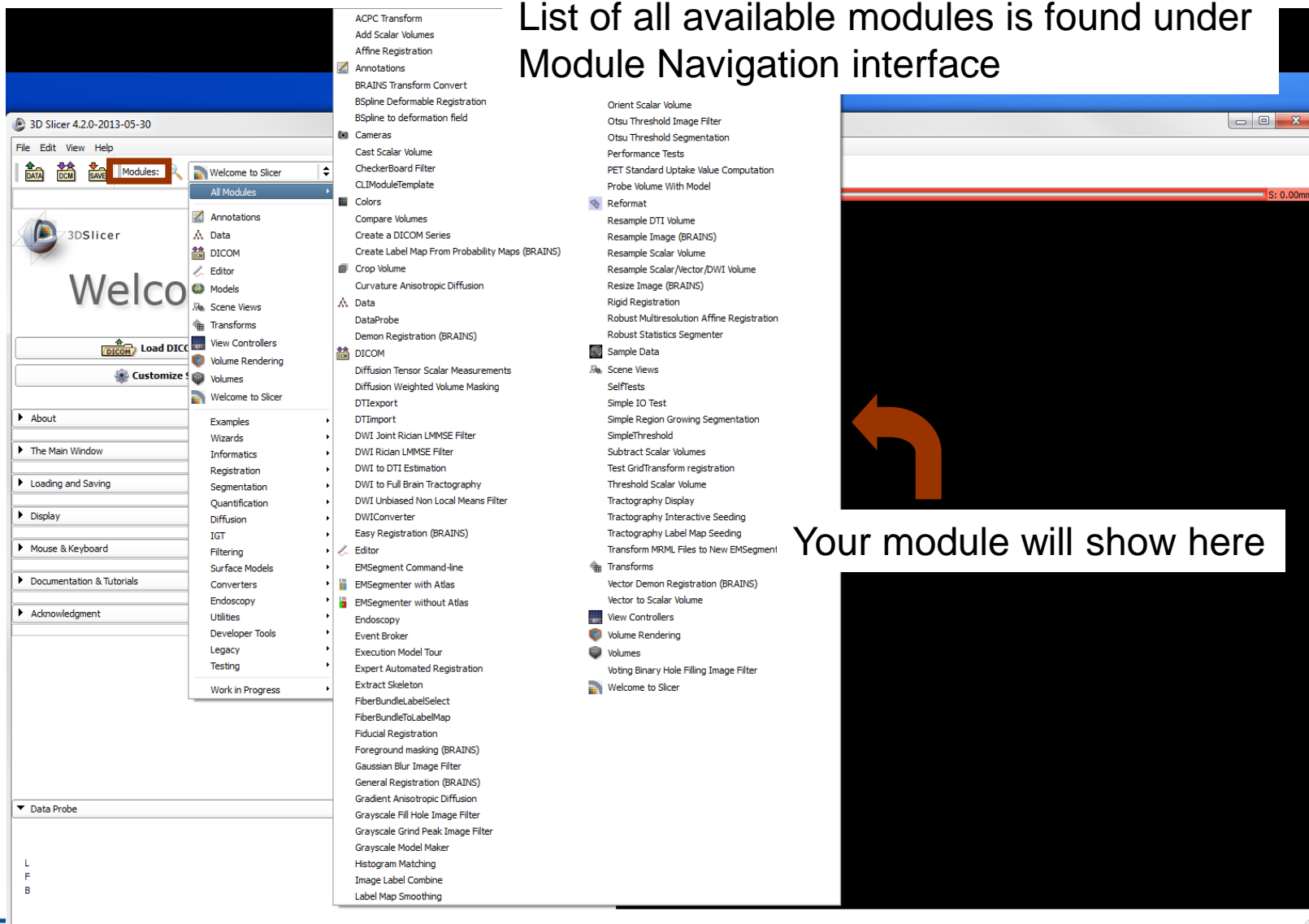- For the general information and "How to" tutorials please visit *http://www.slicer.org/slicerWiki/index.php/Documentation/Nightly/Training*

3

Material

- We recommend to build Slicer 4 from source

- Refer to the following page:
  http://www.slicer.org/slicerWiki/index.php/Documentation/Nightly/

  Developers/Build_Instructions

- Read prerequisites and platform specific instructions and install all required tools

- Checkout Slicer source

- Configure and build

- Become Slicer community member:
  http://www.slicer.org/slicerWiki/index.php/Documentation/Nightly/

  Developers/StartHere

National Alliance for Medical Image Computing
http://www.na-mic.org

4

Run Slicer: *path-to-SlicerSuperbuild/Slicer-build/Slicer.exe*

List of all available modules is found under Module Navigation interface

Your module will show here

CLI module

- Standalone executables, shared libraries or scripts

- Introduced via plugin mechanism

- XML description produces UI

- Command line parsing code

- Link: *http://www.na-mic.org/Wiki/index.php/File:Slicer4_CLI.ppt*

Creating module: Step 1

- Make sure that any version of Python is installed on your computer

- From Slicer source directory run the command:
  ./Utilities/Scripts/ModuleWizard.py --template ./Extensions/Testing/CLIExtensionTemplate --target ../My_Module My_Module

- This command created a new directory "My_Module" parallel to Slicer source directory

Build extension: Windows

- Run cmake

- Set Slicer_DIR to path-to-Slicer-Superbuid/Slicer-build

- Set *../My_Module* as a source directory

- Choose a build directory

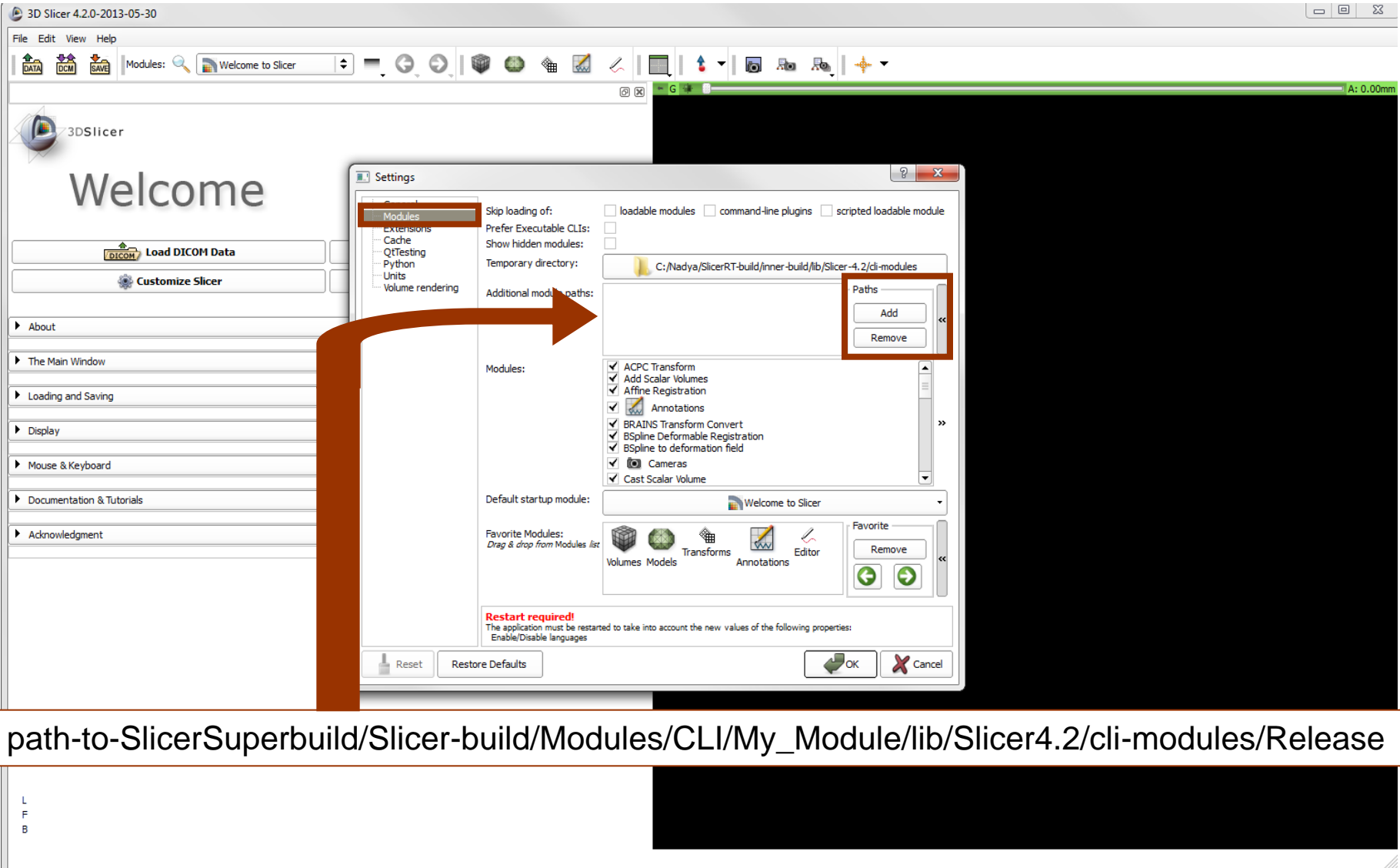- Use default settings

- Compile using VC

Build extension: Linux

- $ mkdir My_Module-build

- $ cd MyModule-build

- $ cmake -DSlicer_DIR:PATH=/path-to-Slicer-Superbuild/Slicer-build ../My_Module

- $ make

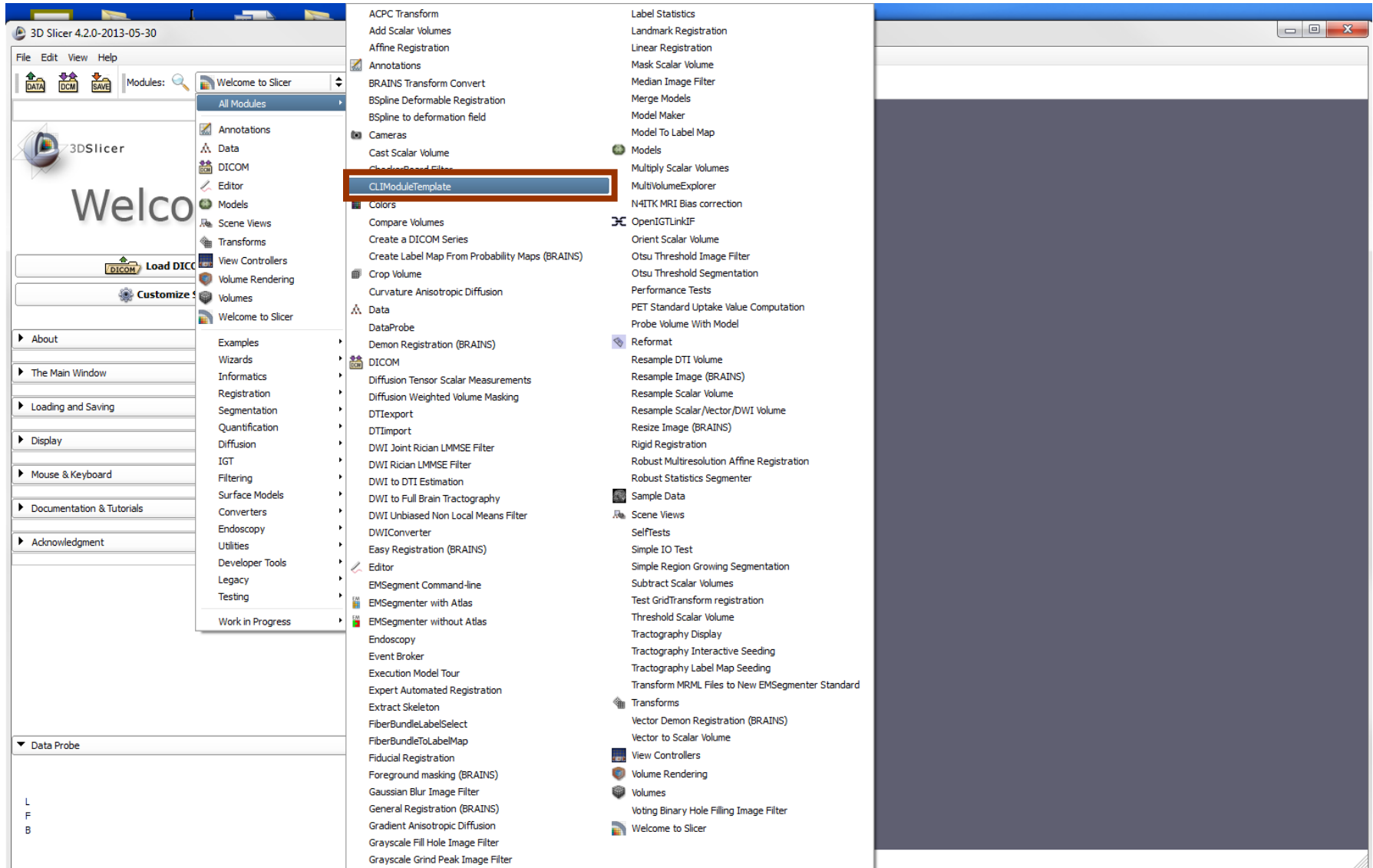# Set the path to My_Module in the Application Settings
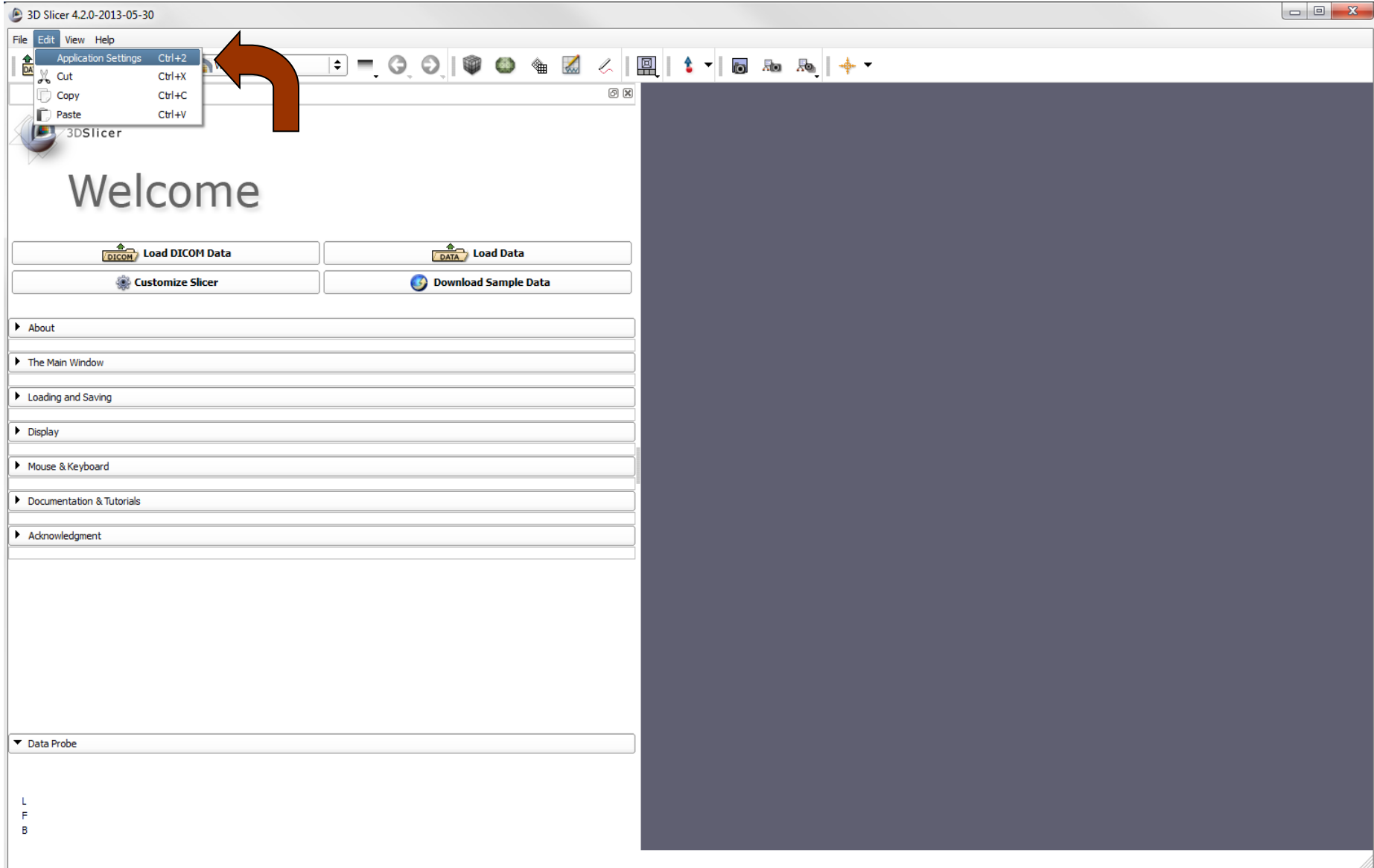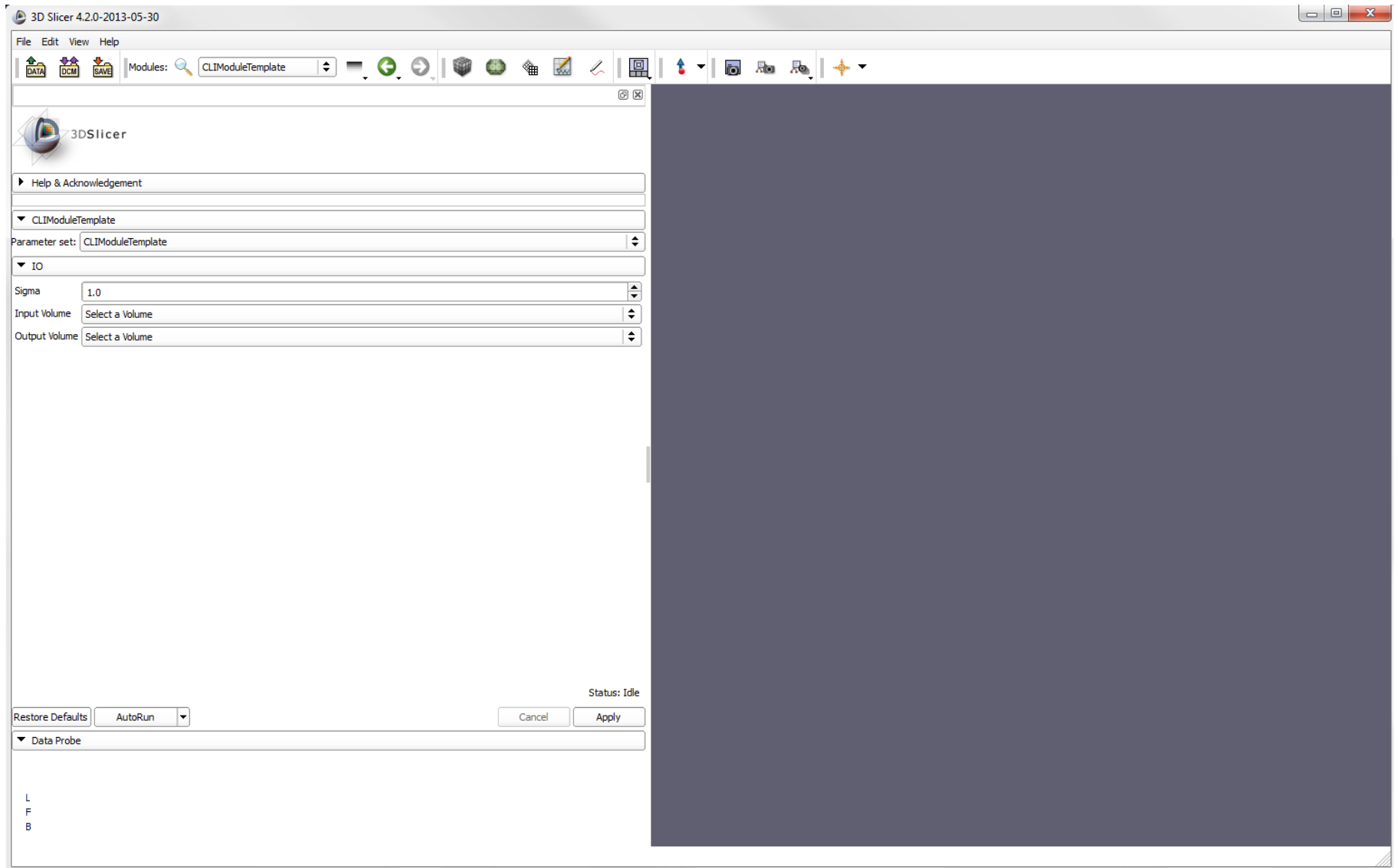
path-to-SlicerSuperbuild/Slicer-build/Modules/CLI/My_Module/lib/Slicer4.2/cli-modules/Release

Restart Slicer!

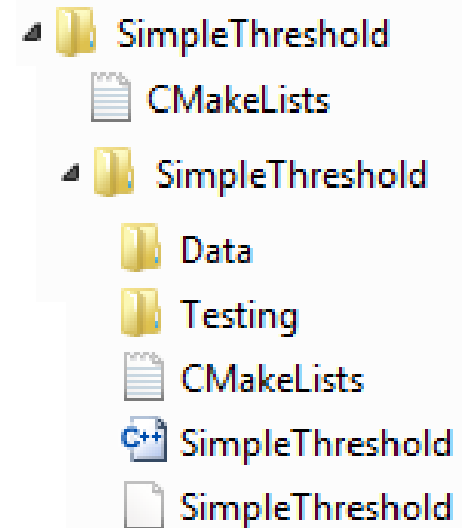# Find the module CLIModuleTemplate in the Module Navigation interface

Open the module. Congratulations!

Creating module: Step 2

- Download sample data:
  https://forge.abcd.harvard.edu/gf/download/

  frsrelease/85/2851/hello_cli.zip
- The name of directory is
  the name of the Module as
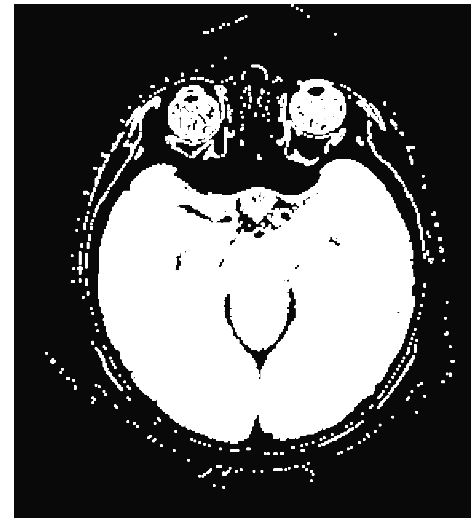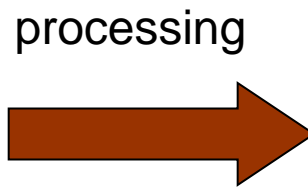 it appears in the list of modules

Module function

processing

Input image

Output image

Module description

GUI

IO

| Input Volume | Select a Volume |
| Output Volume | Select a Volume |

ThresholdParameters

| Lower Threshold | 10 |
| Upper Threshold | 50 |

XML

```
<image>
    <name>inputVolume</name>
    <label>Input Volume</label>
    <channel>input</channel>
    <index>0</index>
    <description><![CDATA[Inputvolume]]></decription>
  </image>
```

```
<integer>
  <name>lowerThreshold</name>
  <longflag>--lowerThreshold</longflag>
  <description><![CDATA[The lower threshold]]></description>
  <label>Lower Threshold</label>
  <default>10</default>
 </integer>
 <integer>
```

Module coding

```
typedef itk::ImageFileReader<InputImageType>  ReaderType;
typedef itk::ImageFileWriter<OutputImageType> WriterType;
typename ReaderType::Pointer reader = ReaderType::New();
reader->SetFileName( inputVolume.c_str() );
 itk::GetImageType(inputVolume, pixelType, componentType);


typedef itk::BinaryThresholdImageFilter<
   InputImageType, OutputImageType>  FilterType;
typename FilterType::Pointer filter = FilterType::New();
filter->SetLowerThreshold(lowerThreshold);
```

Compile the module

- From Slicer source directory run the command: ./Utilities/Scripts/ModuleWizard.py --template ../SimpleThreshold --target ../SimpleThreshold SimpleThreshold

- Build extension (pp 9,10)

- Start Slicer

- Set the path to SimpleThreshold in the Application Settings (pp 11,12)

- Restart Slicer

Loading data

Navigate to the location of ct_head.mha

Click OK

Choose ct_head.mha in Input Volume and "Create new Volume" in Output Volume

Click Apply

Saving data

In "Save" menu choose volume and format to save

Click Save

Contact information

If you have questions please contact Greg Sharp at
E-mail: gcsharp@partners.org
Tel: (617) 724 3866

Acknowledgments

National Alliance for Medical Image Computing

NIH U54EB005149

National Alliance for Medical Image Computing
http://www.na-mic.org