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
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
Material

- We recommend to build Slicer 4 from source
- Refer to the following page:
- Read prerequisites and platform specific instructions and install all required tools
- Checkout Slicer source
- Configure and build
- Become Slicer community member:
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
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-Superbuild/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 My_Module-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

Input image  

processing  

Output image
Module description

**GUI**

- **Input Volume**
  - Select a Volume

- **Output Volume**
  - Select a Volume

**XML**

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

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

```xml
<integer>
  <name>upperThreshold</name>
  <longflag>--upperThreshold</longflag>
  <description><![CDATA[The upper threshold]]></description>
  <label>Upper Threshold</label>
  <default>50</default>
</integer>
```
typedef itk::ImageFileReader<InputImageType> ReaderType;
typedef itk::ImageFileWriter<OutputImageType> WriterType;
typename ReaderType::Pointer reader = ReaderType::New();
reader->SetFileName(inputVolume.c_str());
#define 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 Institutes of Health
NIH / NCI 6-PO1 CA 21239
Federal share of program income earned by MGH on C06CA059267

Progetto Rocca Foundation
A collaboration between MIT and Politecnico di Milano