Intro to Doxygen

Stephen Herbener

JEDI Core Team

4/19/18

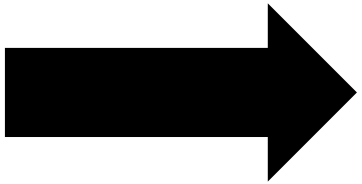
Doxywizard

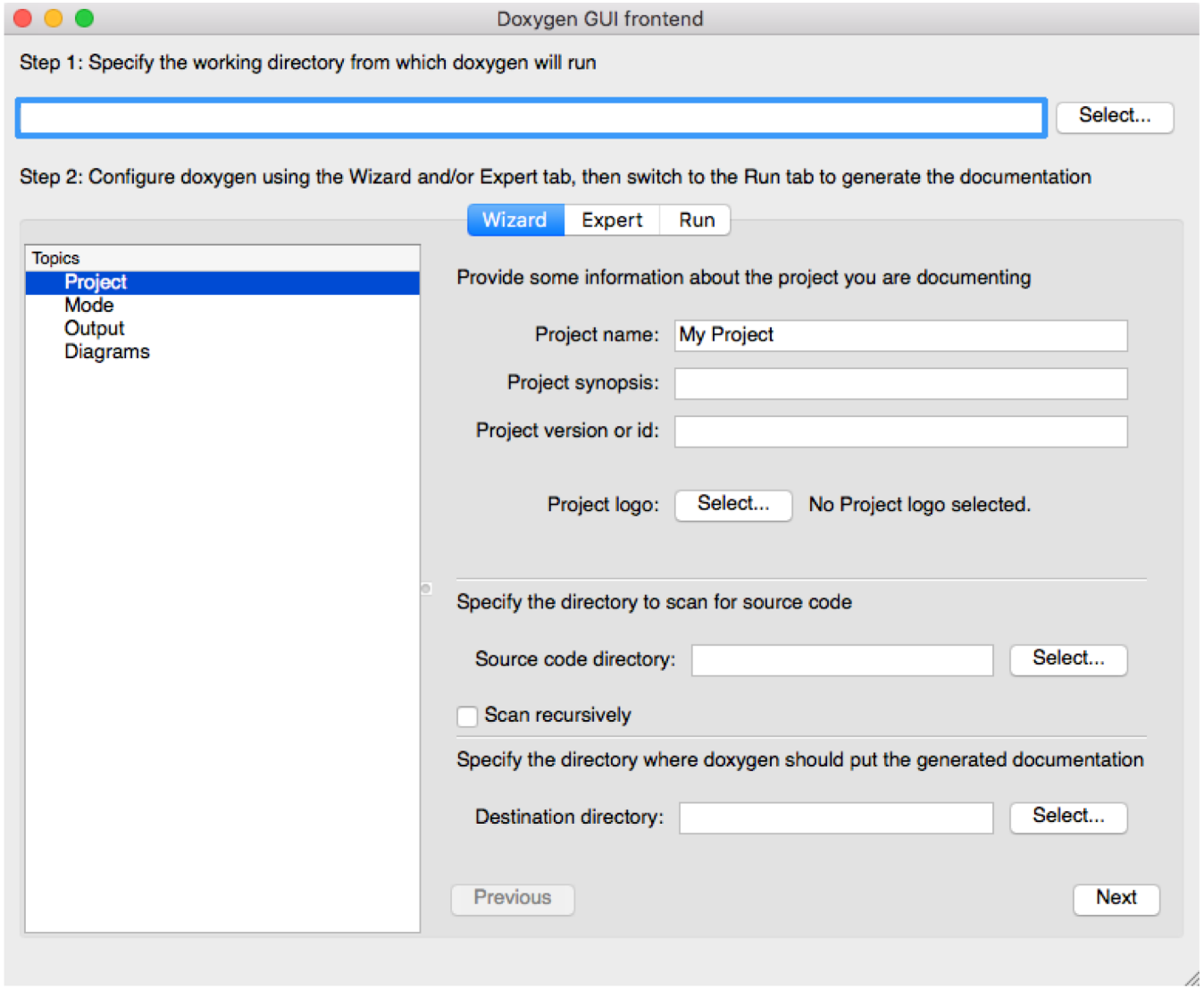
- GUI that helps you configure and run doxygen
- Doxywizard assists in the creation of a doxygen configuration file
  - User enters information through GUI forms
  - The configuration file created by Doxywizard can be used directly by doxygen
    - Enables batch processing from the command line:

      doxygen <config_file>
- Doxywizard can run doxygen for you
  - Hit the “Run” button
  - Captures output from doxygen in a GUI window
- Doxywizard is supported by the developers of doxygen
  - https://www.stack.nl/~dimitri/doxygen/manual/doxywizard_usage.html

Doxywizard: Start up

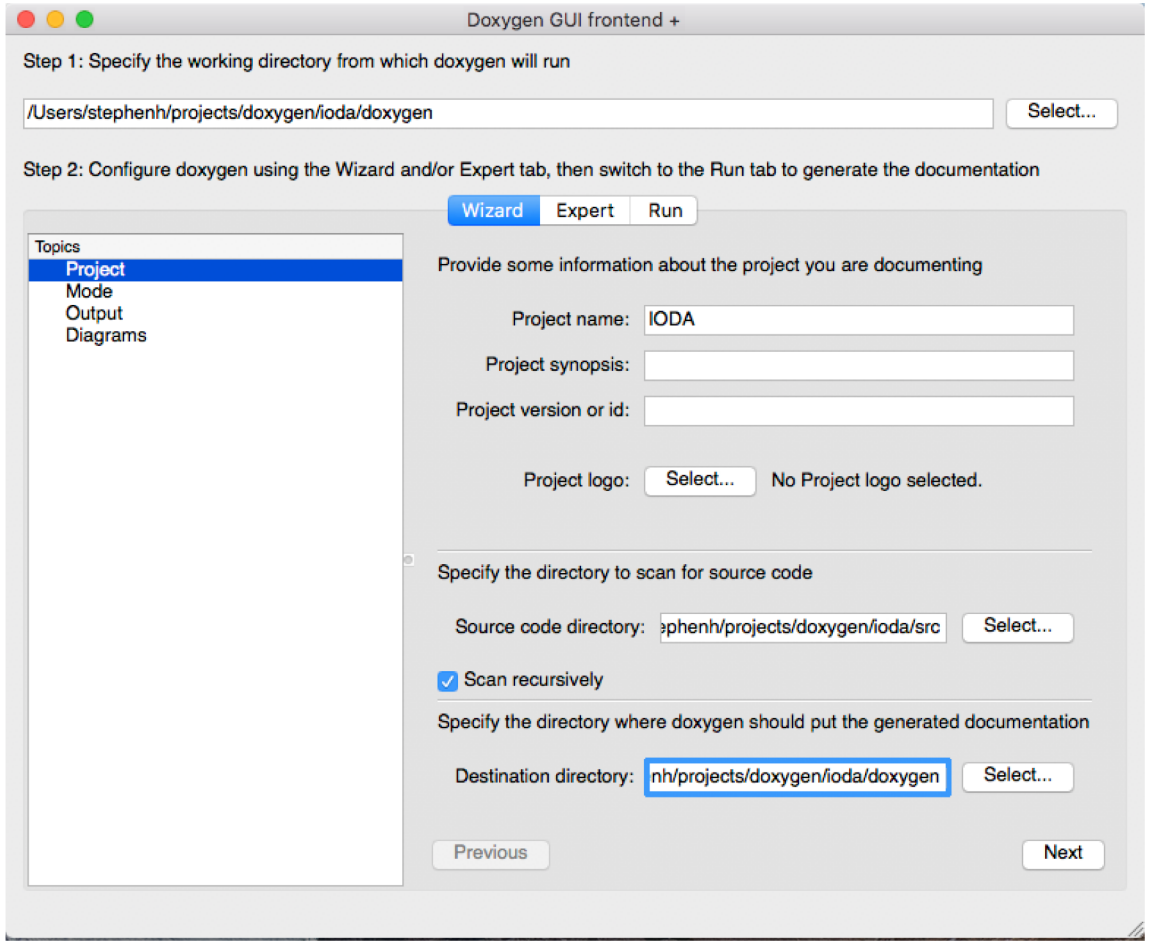
On the Mac, click on the Doxygen icon in the Applications folder



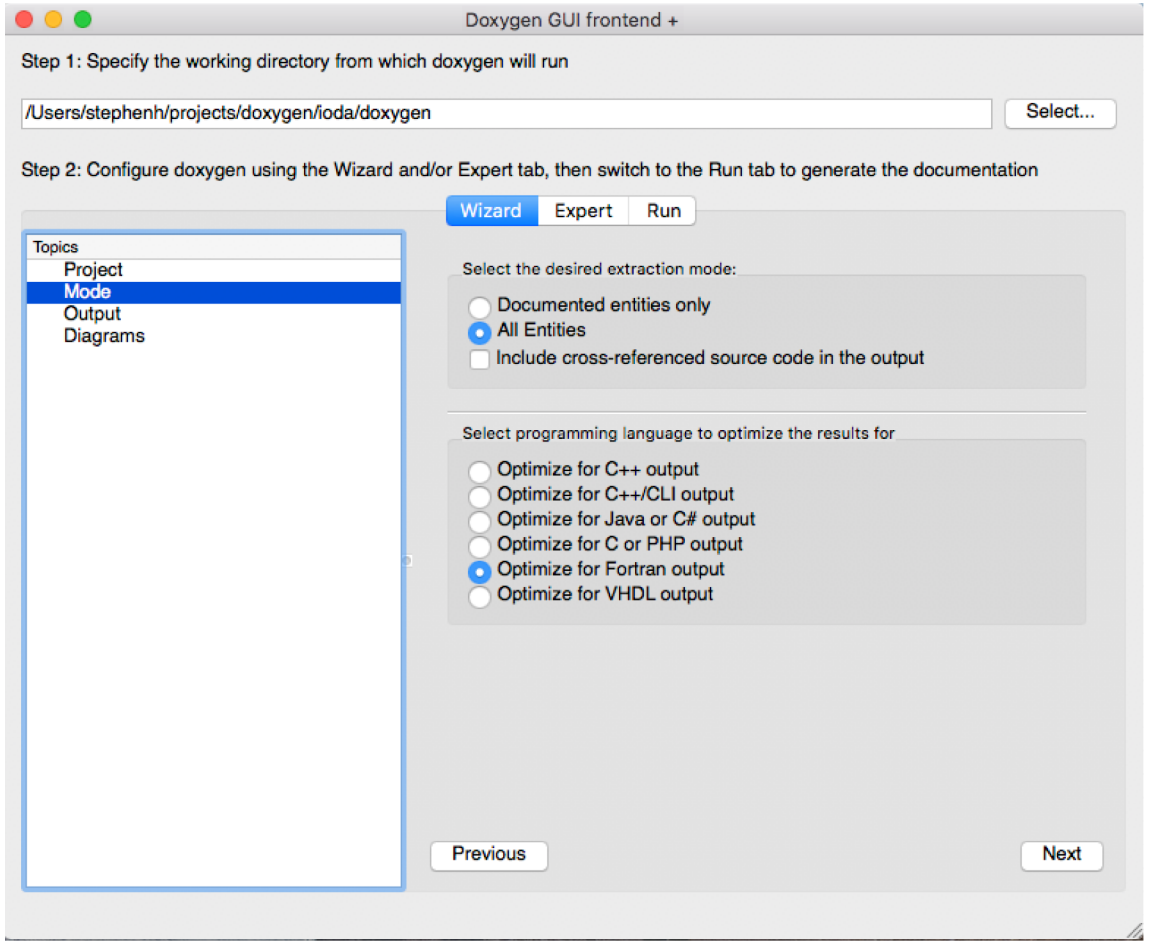


- Configuration buttons
  - Wizard: Quick and easy
  - Expert: All the gory details

Doxywizard: Wizard configuration

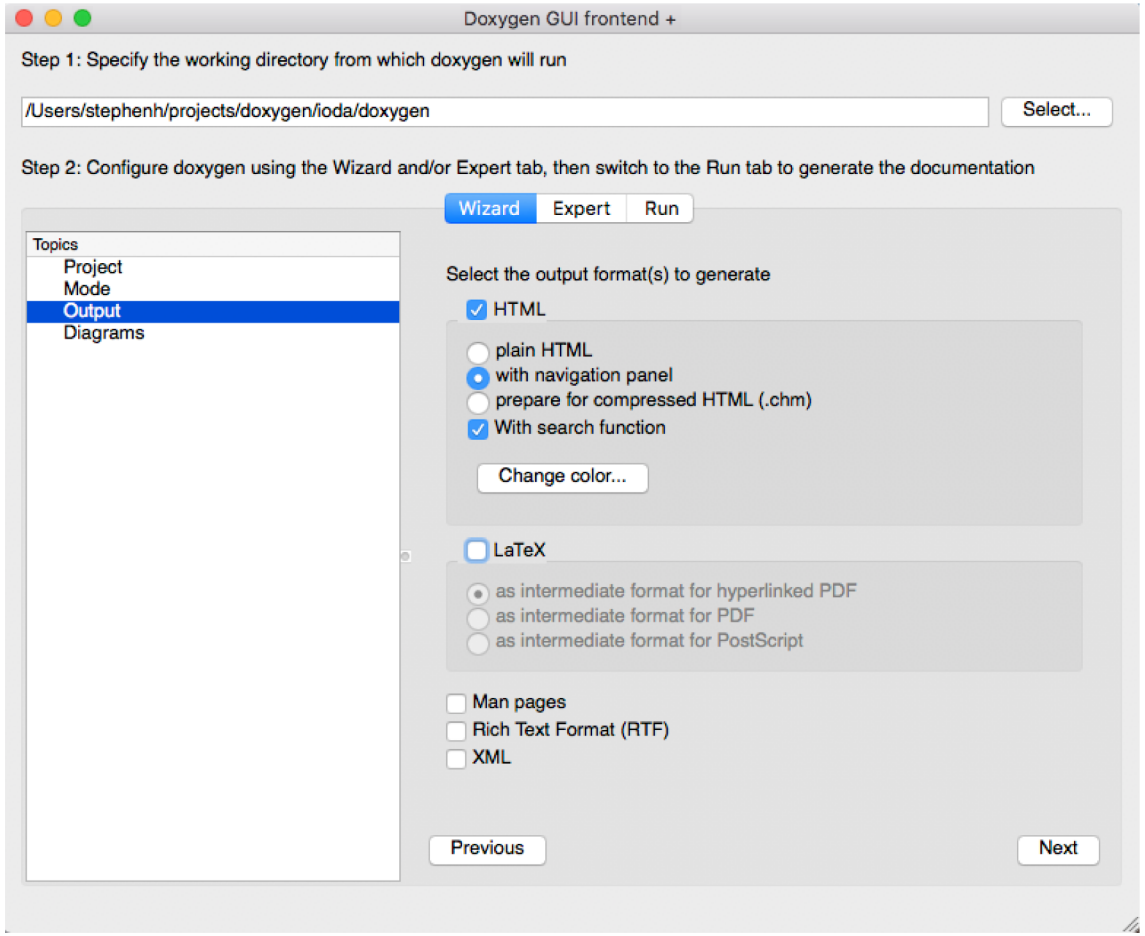


- Project
  - Set paths to source code and destination to output documentation

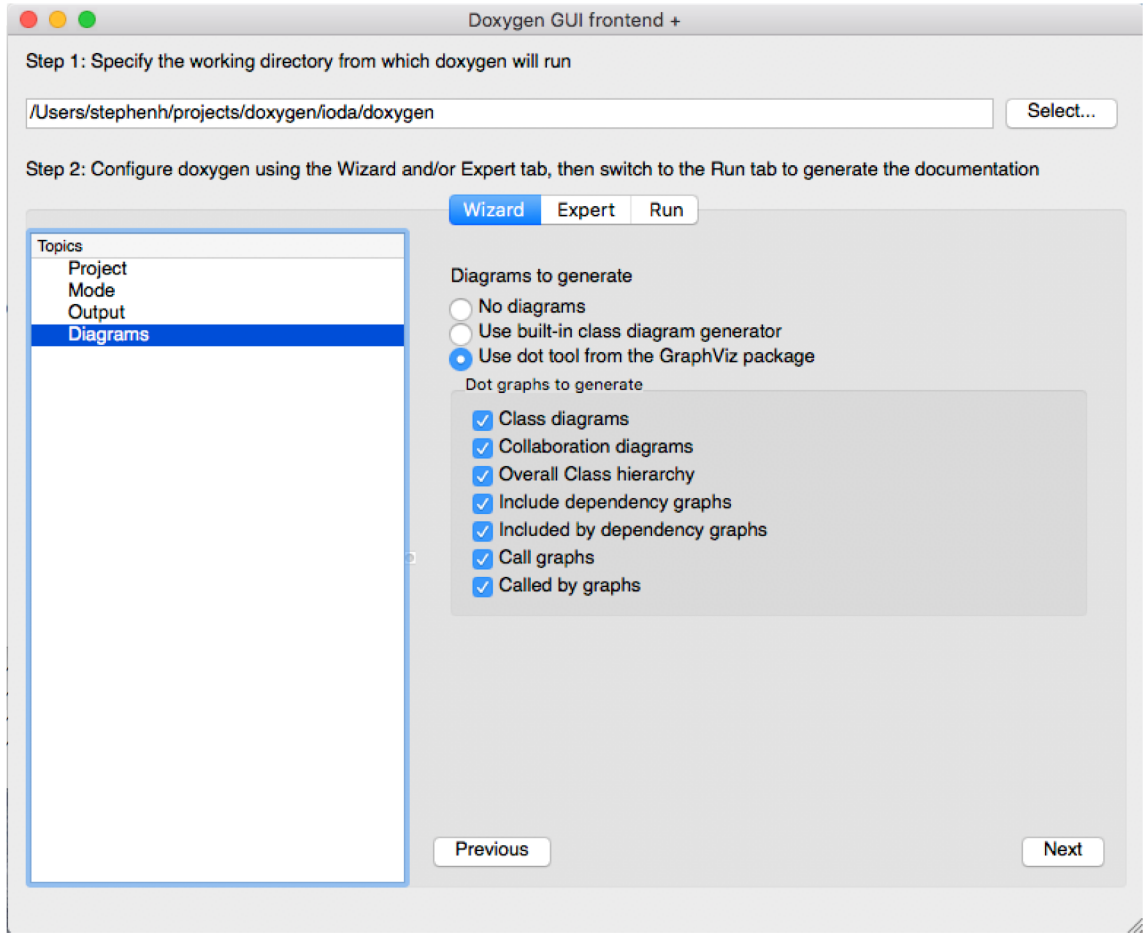


- Mode
  - Select what to extract and the primary programming language in the source code

Doxywizard: Wizard configuration

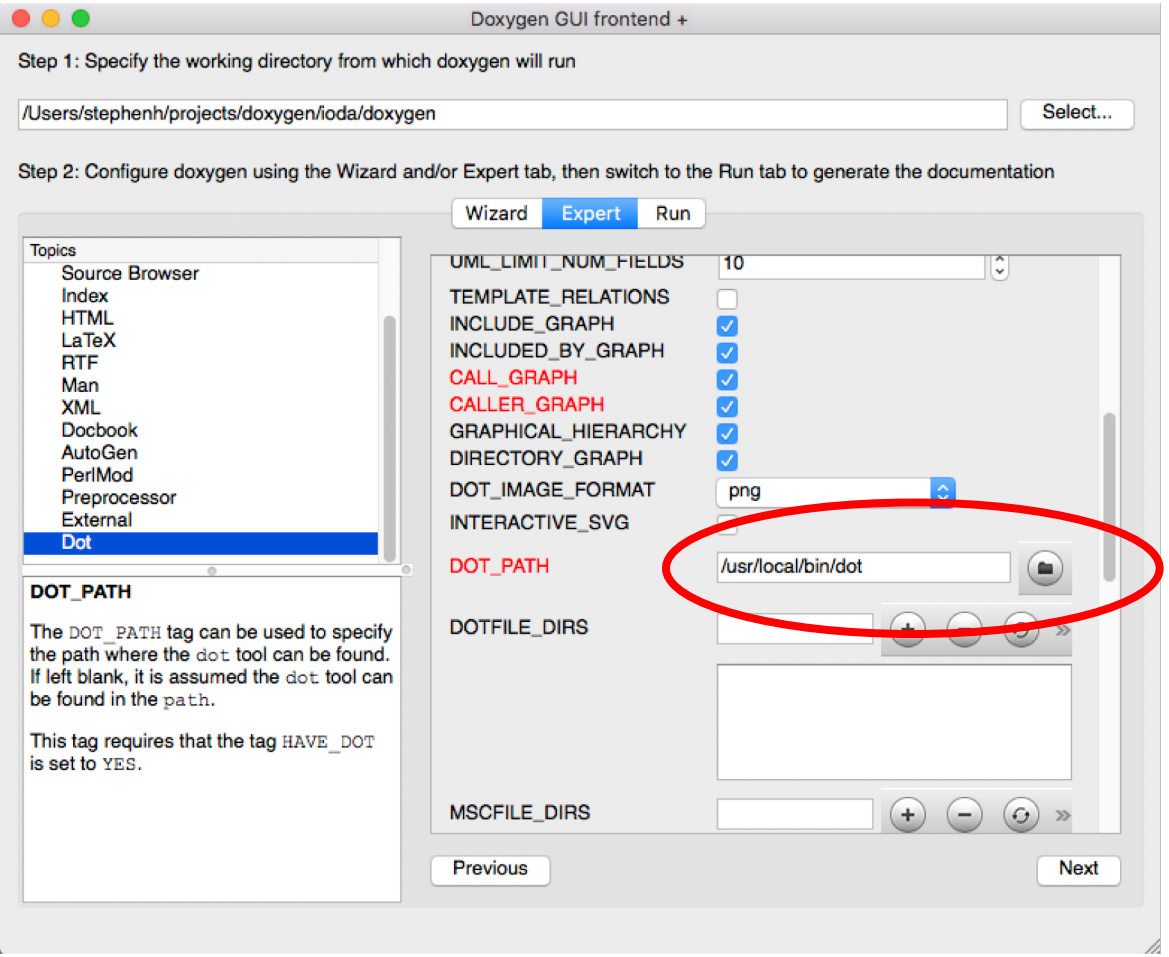


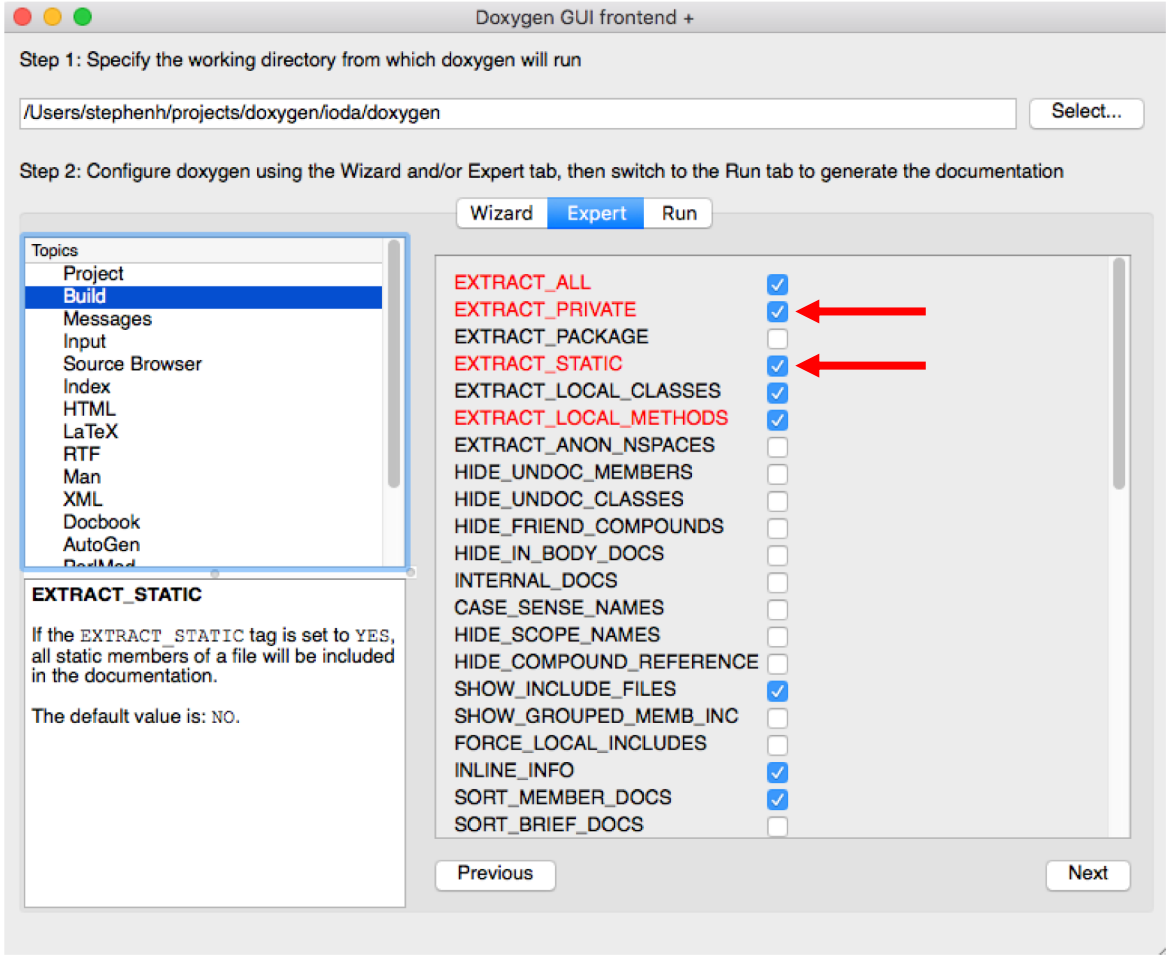
- Output
  - Set the formats for the generated documentation



- Diagrams
  - Select any diagrams to be placed in the generated documentation

Doxywizard: Expert configuration

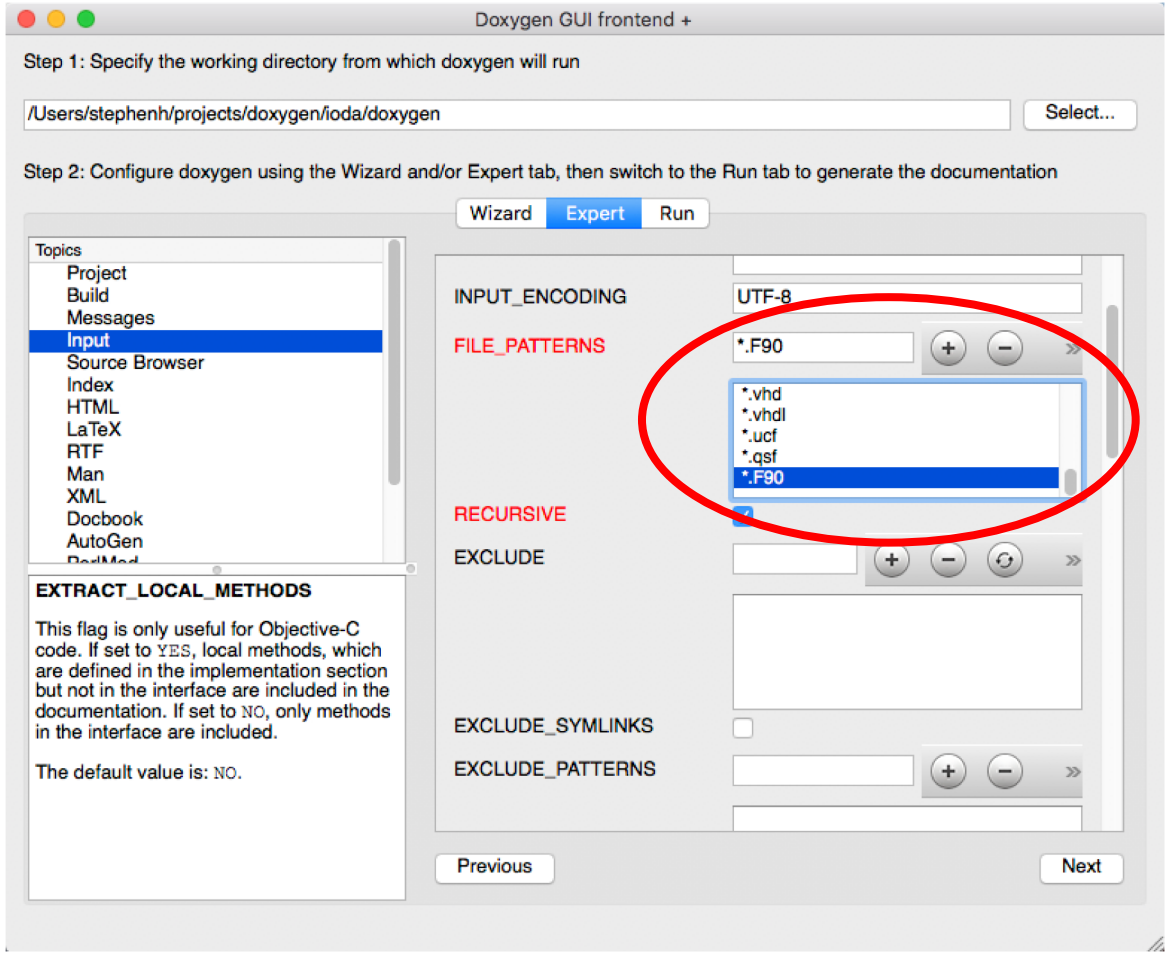




- Set the path to the dot executable
  - Typically: /usr/local/bin/dot

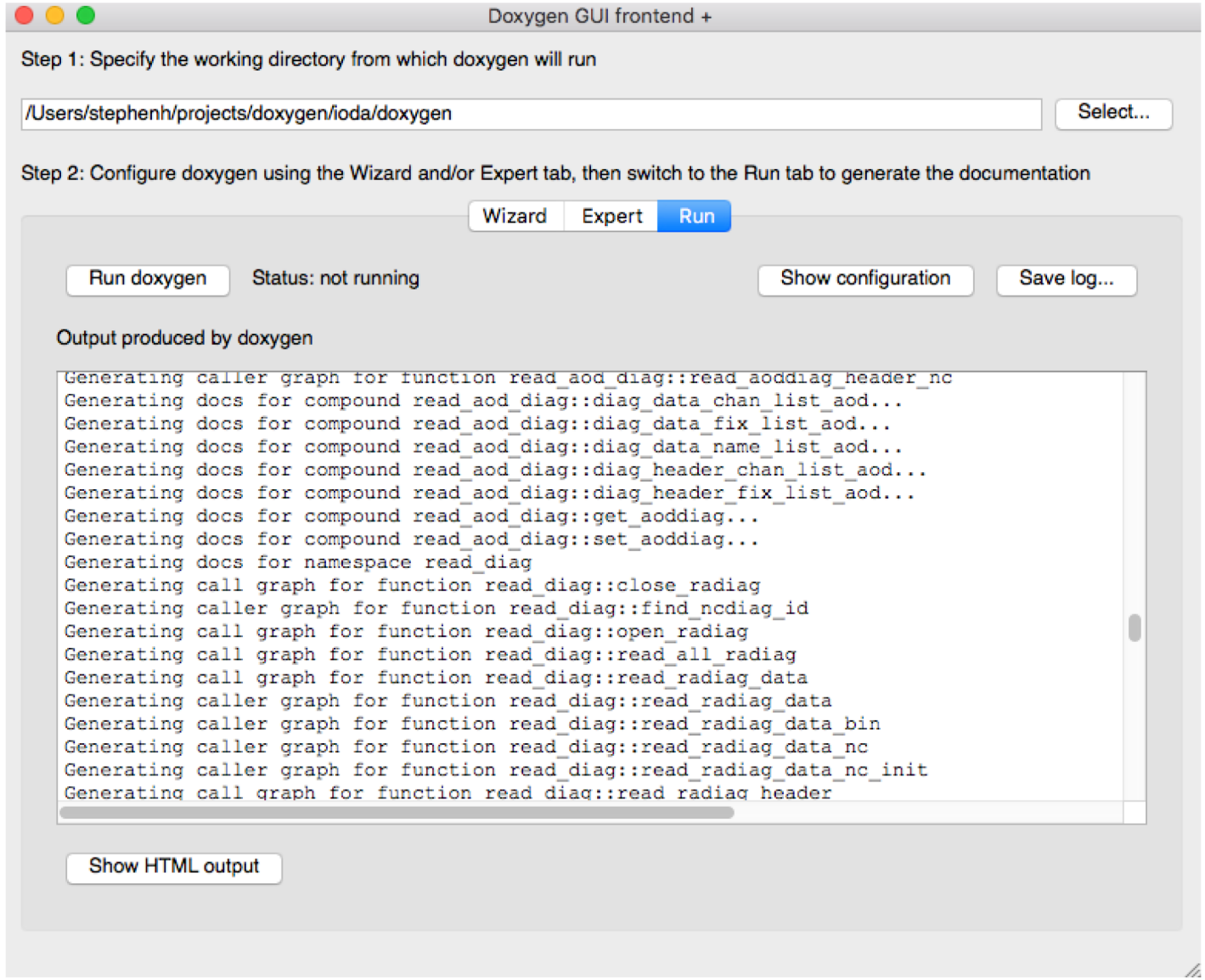
- EXTRACT_PRIVATE will include private data members and methods in generated documentation
- EXTRACT_STATIC will include static members in generated documentation

Doxywizard: Expert configuration



- Make sure to include *.F90 file pattern

Doxywizard: Run doxygen



- You will get the same result by running on the command line:

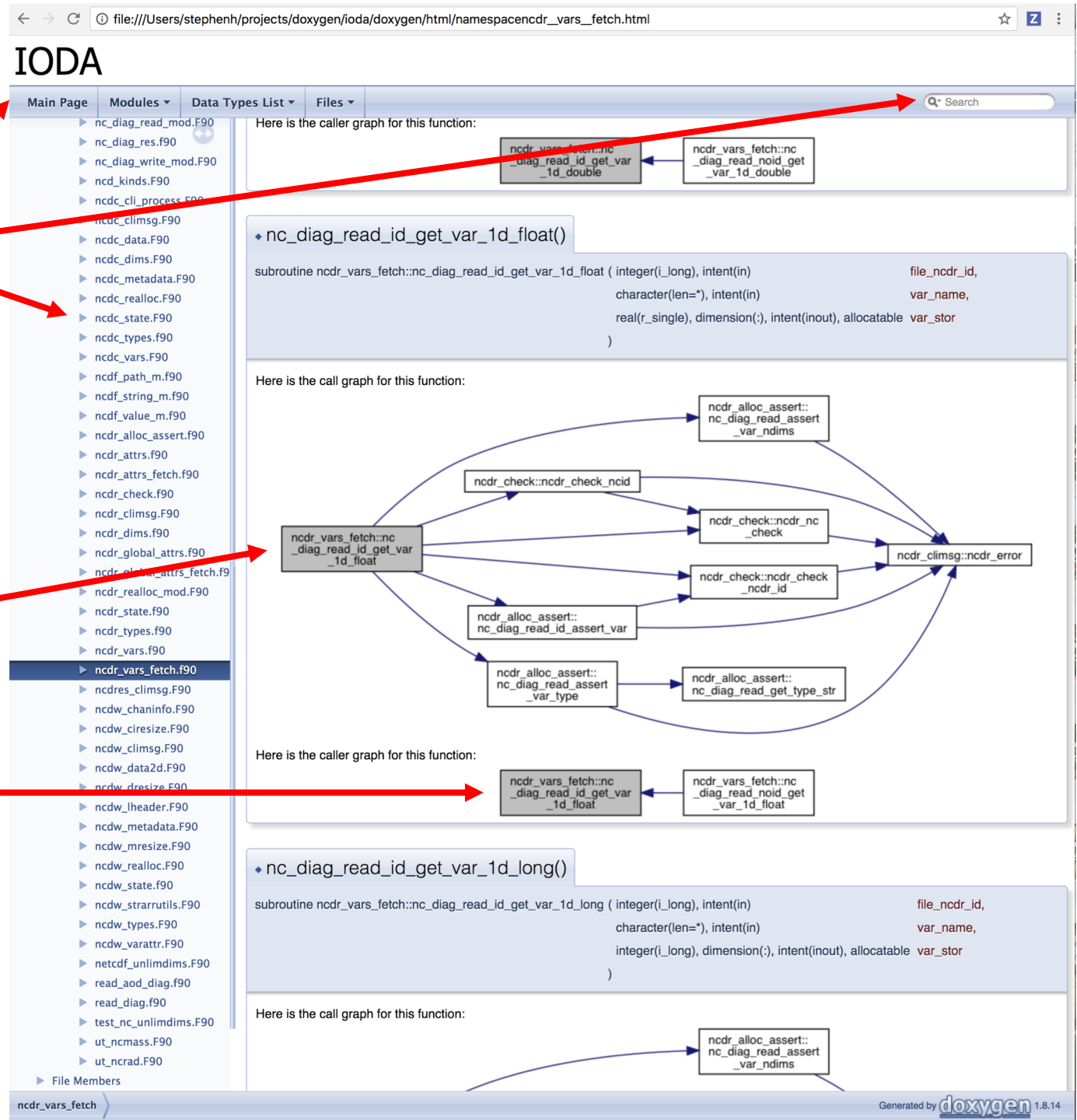
  doxygen Doxyfile

Doxywizard: HTML output

Navigation

Call graph:
Who this routine calls

Caller graph:
Who calls this routine



Getting organized

- Code writers
  - Place appropriate comments in source files for Doxygen
  - Test in your local feature branch using Doxywizard
- Core team
  - Construct a flow that visits all the repositories, runs doxygen and publishes the generated documentation
  - This flow would pull from GitHub and operate on:
    - develop branch
      - Test the documentation generation flow
    - master branch, using release tags
      - Publish documentation that is in sync with releases
- We may want to create a directory in a repository, as needed, to hold the source for any relevant documentation that can’t be generated from the source code.
  - Doxygen
  - Sphinx
  - OOPS repository has an example of this
    - Subdirectory: Documents
    - Description of OOPS (doxygen), manual for QG model (LaTex), etc.