Documentation of Analysis Code
Analysis Procedures
and more.

Benedikt Zihlmann

December 5, 2021

Documentation options

- Documentation is essential
- There are several "layers" of documentation
  - Basic code documentation: comments, doxygen
  - Basic analysis steps documentation: pdf, ...
  - Specific "package" documentations: pdf, ... (ReactionFilter, DSelector, AmpTools, ...)
  - How to pages, FAQ pages: wiki

Current Status

Wiki:

- At top level wiki page no link labled "Documenation"
- All Documentation is burried at various levels of different links
- Hard to find if you do not already know where to look

→ Task1: Coherency, bring order into chaos.

Topic Documentation:

DSelector `https://www.overleaf.com/project/5ca37385d3a0d471b3c1a430`

AmpTools `https://github.com/mashephe/AmpTools/wiki`
`https://github.com/mashephe/AmpTools/blob/master/AmpTools_User_Guide.pdf`

Overview `https://www.overleaf.com/9649723516hgfwhsnwdjvw`

→ Task2: Updates, by authors AND users.

Code Documentation: doxygen (halld_recon, halld_sim)
→ Task3: Missing in most DANA plugins.

What is doxygen

Doxygen is a "frame work" that geneates documentation in web/html form based on "marked commets" found in source code like C++ or python or ... .
Doxygen uses a configuration file to define its behavior to produce:

- A list of all code files, a page for each header and code file, navigable
- A list of all classes found, naviable. Each page represents one class and include class parameters, public (and private) member functions and variables.
- Behavior of doxygen is configurable with requried config file

Example: doxygen MyConfigFile

Doxygen Config file

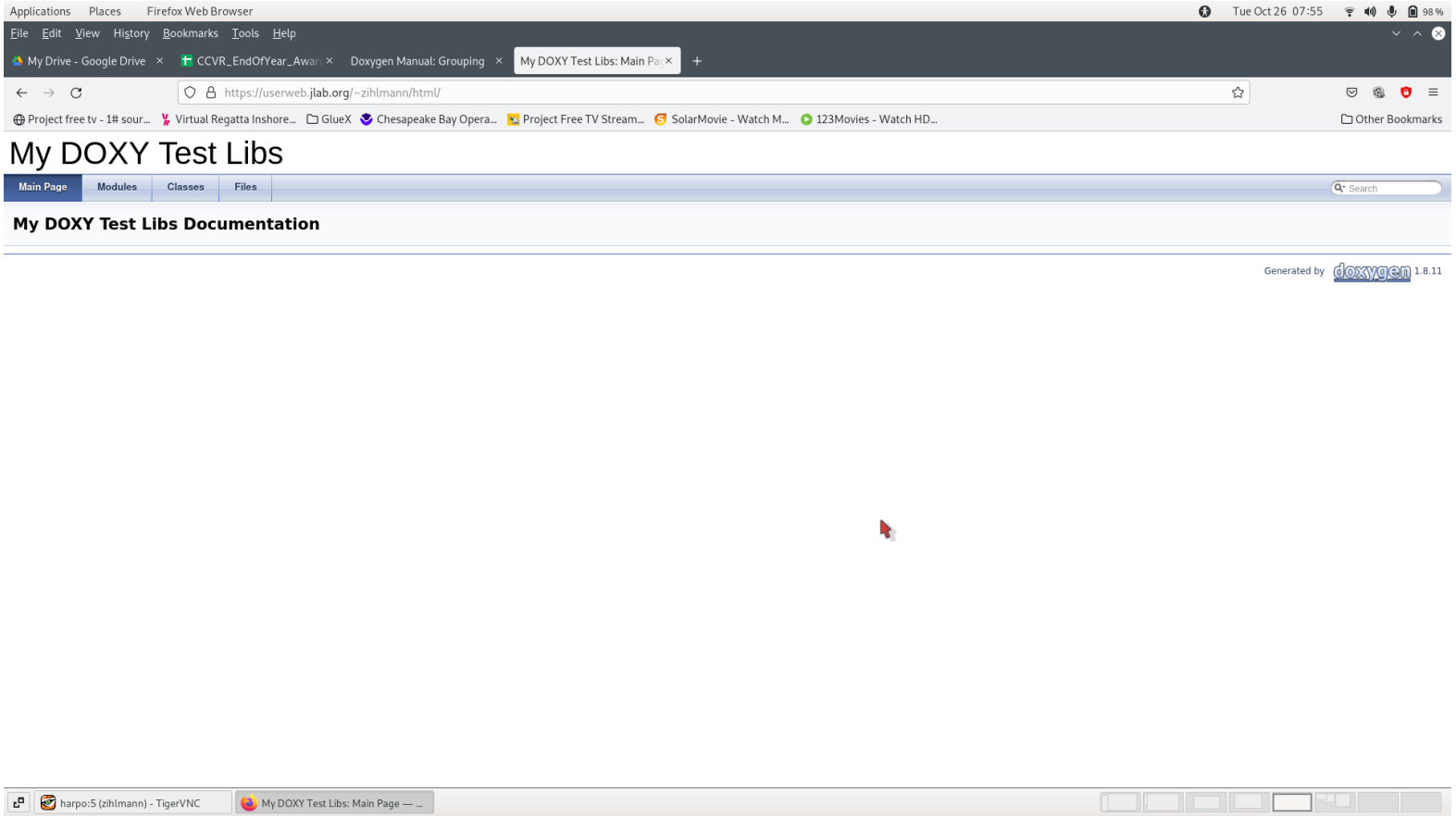
Example of some configuration parameters:

- OUTPUT_DIRECTORY = locationWhereToWriteHTML
- BRIEF_MEMBER_DESC = yes # preceed detailed doc by brief command
- REPEAT_BRIEF = yes # prepend brief description
- AUTOLINK_SUPPORT = yes # identify known strings and generate links
- EXTRACT_PRIVATE = yes # include private members in documentation
- EXTRACT_LOCAL_CLASSES = yes # include locally defined classes
- RECURSIVE = yes # search subdirectories for code
- FILE_PATTERNS = # which files to scan for content

and many, many, many, many more.....

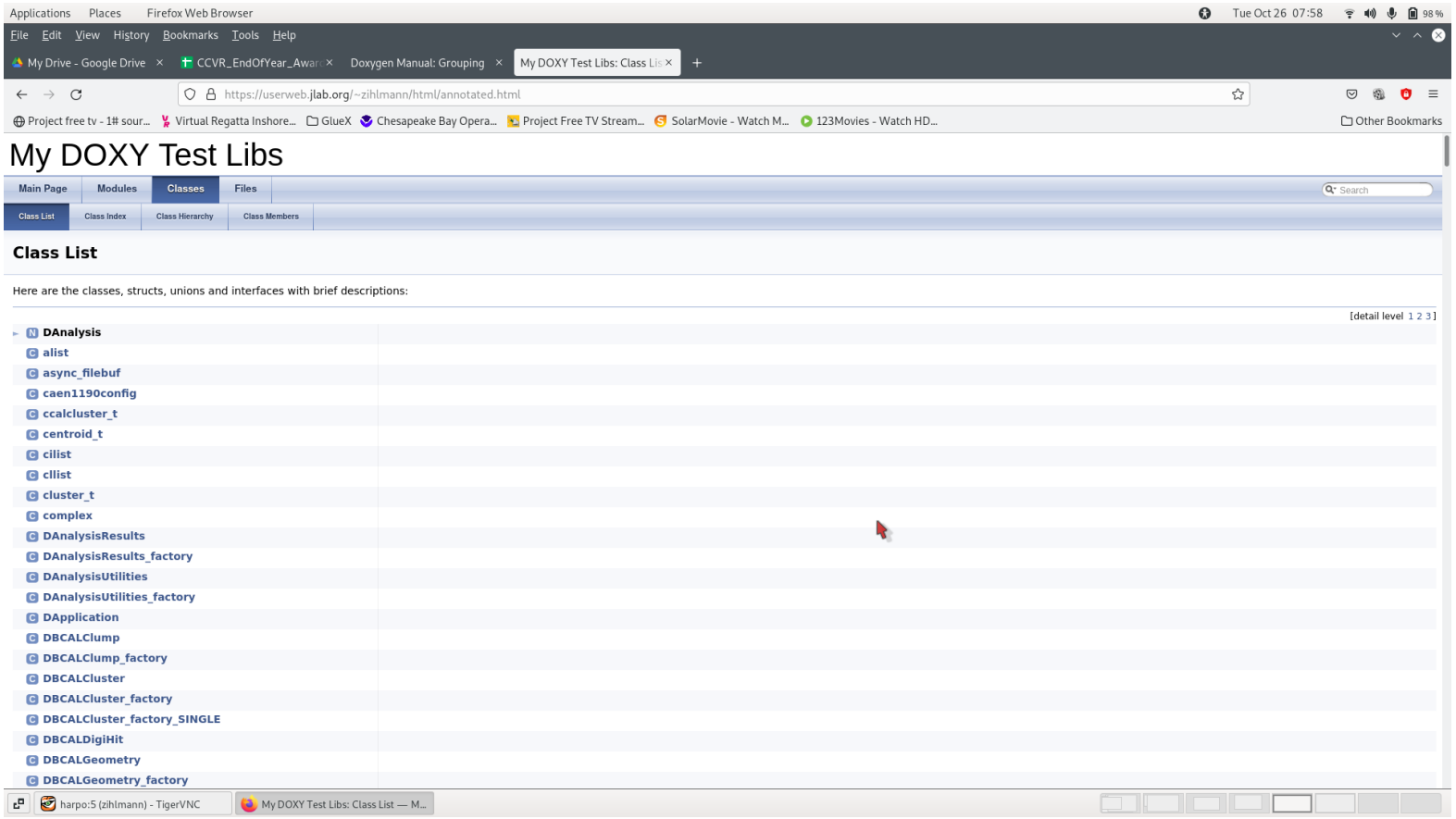
Example doxygen result

Example of test: Overview page



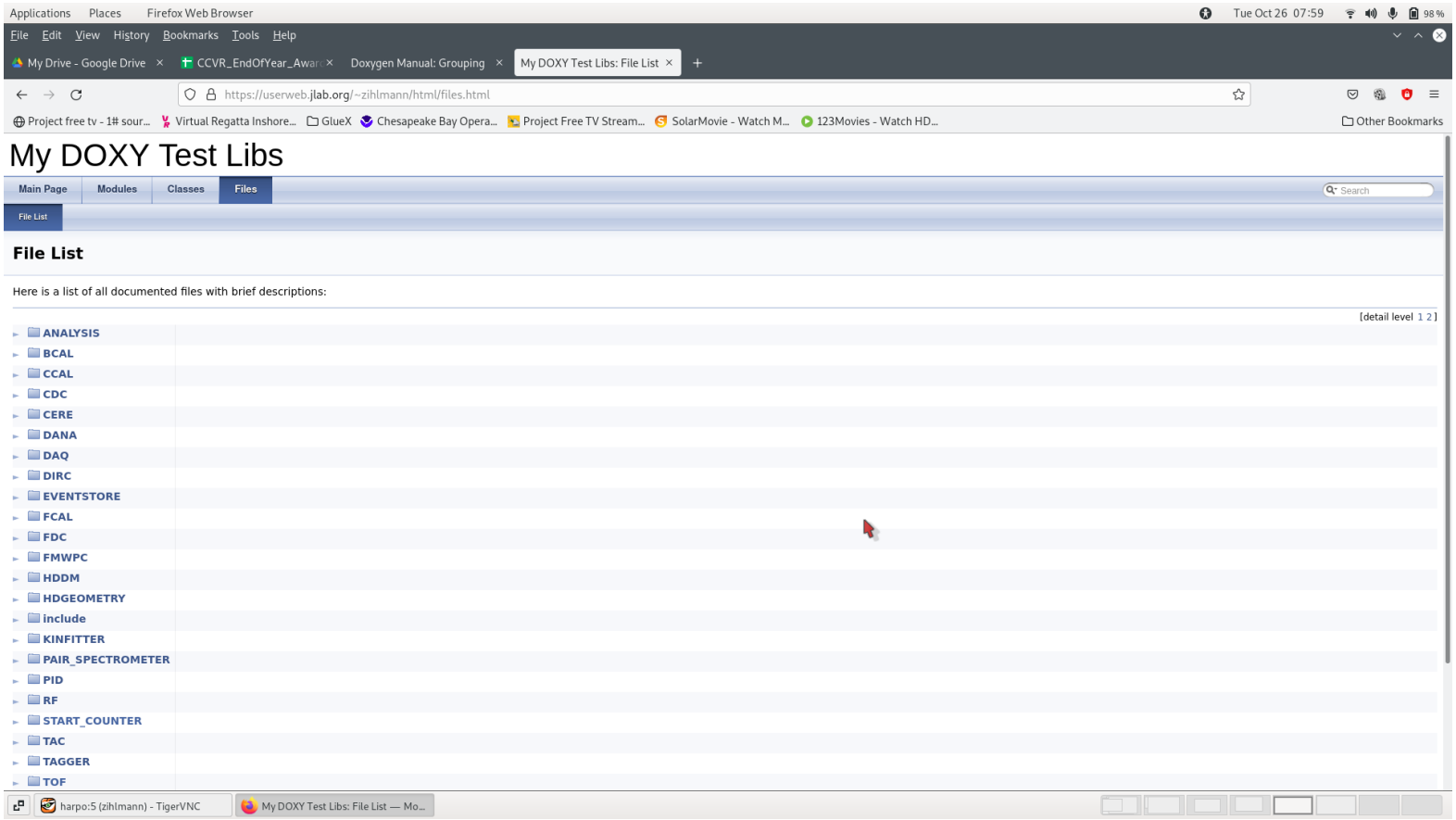
Example doxygen result

Example of test: Class List (dir if inheritance)

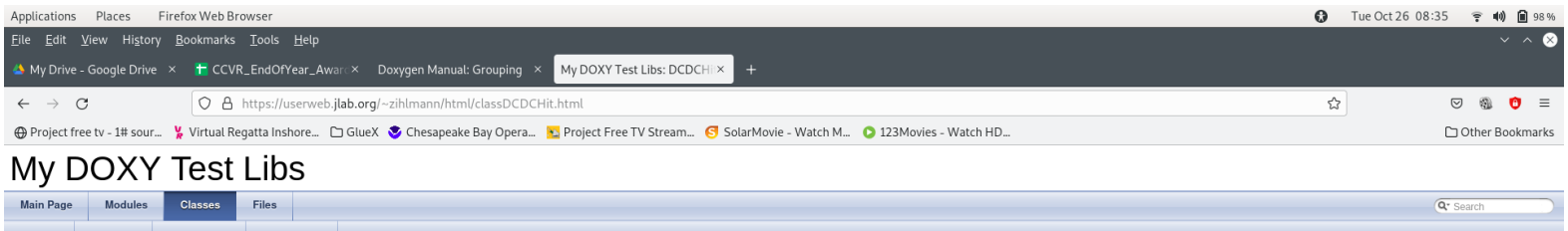


Example doxygen result

Example of test: File List, in subdirectories

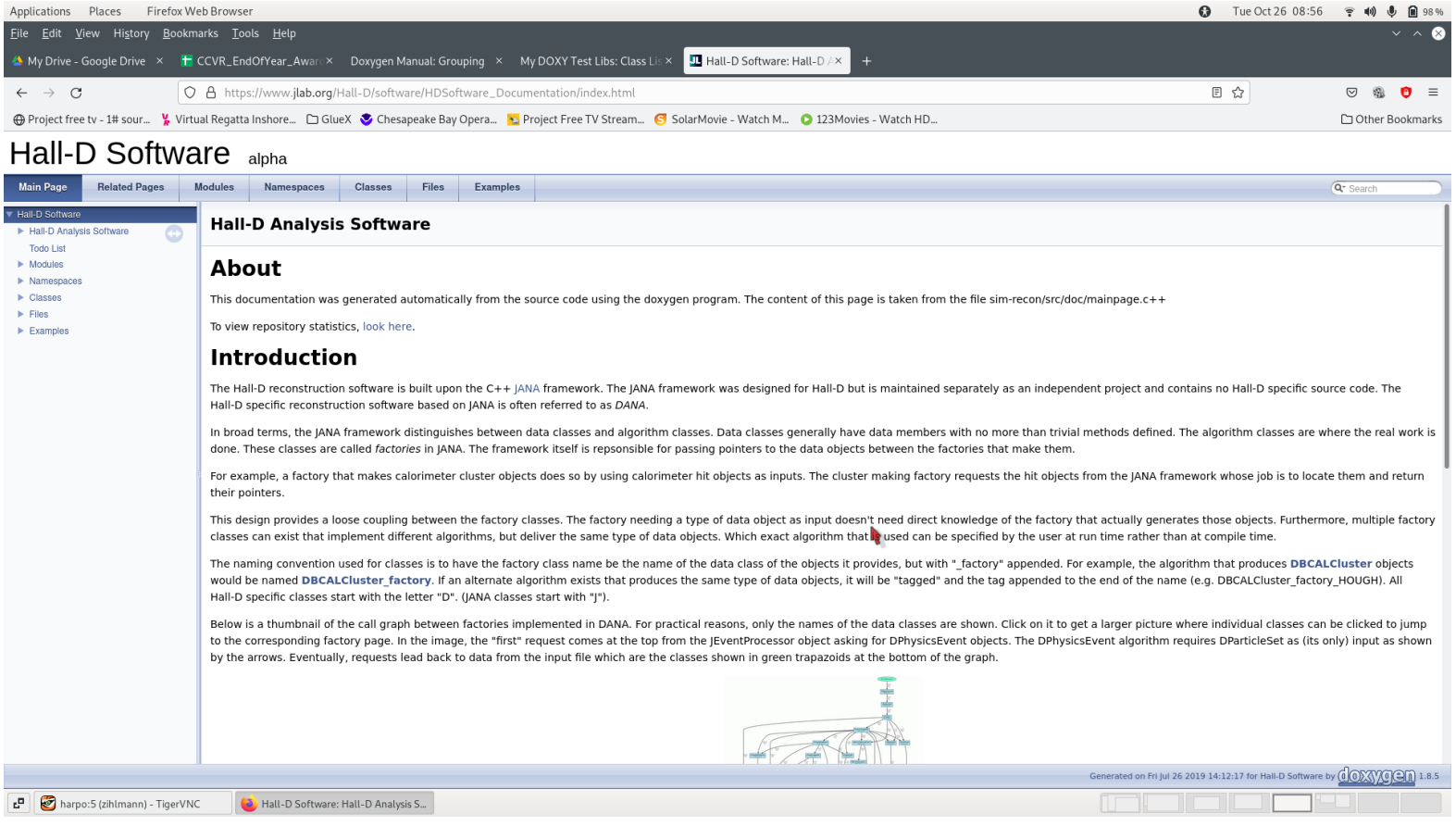


Example of DCDCHit container class:



Current doxygen GlueX page

config file: "halld_recon/src/doc/Doxyfile"



Usage/consuming output of Doxygen

Doxygen is usful to explore the capabilities and dependences of code like classes, variables and their methods.

- What is this library about (this = TOF, BCAL, ...)?
- What does this class do?
- What data does this class use/handle/provide?
- What external parameters/data does this class need?
- Could also create man pages if desired.

If you are new, need to learn basic concepts first, you need basic how to documentation. (Doxygen is not useful)

- What is a plugin and how does it work? usage of MakePlugin, ect.
- How to analyze rest files? usage of MakeDSelector, etc.
- What are the capabilities of DSelector?
- How to pages are a good start.(find/search?)

Conclusion (mine)

- Documentation is a two line project with 3 parts.
- Part One: Doxygen to document the actual code.
- Part Two: PDF/text to document(s) the "usage" of code, how to(s)
- Part Three: PDF/text to document(s) concepts of analyses.