

EIC LXR Code Browser

Shuwei Ye, Alexander Undrus

EIC SW Meeting
October 21, 2020

What is LXR

- ❖ LXR provides code browsing/searching on web-browser
 - ✓ Identifier definition/usage search
 - ✓ Free text search
 - ✓ Version comparison
- ❖ LXR makes two separate indexing's:
 - ✓ Identifier/Reference indexing via ctags
 - ✓ Text indexing via glimpse(our choice) or swish
- ❖ LXR output is **dynamic**, the html is generated on-the-fly
- ❖ LXR is written in Perl

LXR Service at BNL

- ❖ BNL provides LXR code browsing services for ATLAS collaboration for ~ 15 years
 - ✓ Managers: Alex Undrus, Shuwei Ye
 - ✓ See Alex's NPPS presentation for details
- ❖ 6 git branches (daily updates) and selected stable releases
 - <https://acode-browser1.usatlas.bnl.gov>
- ❖ EIC LXR code browser is recently created
 - <https://eic-code-browser.sdcc.bnl.gov>
 - ✓ Most recent LXR version 2.3.5
 - ✓ EIC git repository, master branch
 - Same packages as at EIC Doxygen
 - ✓ Daily code update/indexing
 - ✓ With improvements and optimizations

Main Interface of EIC Code Browser



There are 4 main functions

- ◆ **Source navigation**
- ◆ **Diff markup**
- ◆ **Identifier search**
- ◆ **General search**

Source Navigation



EIC code displayed by LXR

[master/fun4all_acts/Tests/](#)

The directories/files can be browsed/clicked by:

- ✓ Top full path (each subdir is clickable)
- ✓ List of content

Last indexation completed on 2020-10-12 07:00:09

	Name	Size	Date (UTC)	Last indexed
	Parent directory	-	2020-10-12 07:00:27	
	Benchmarks/	-	2020-10-12 07:00:09	
	CommonHelpers/	-	2020-10-12 07:00:09	
	Data/	-	2020-10-12 07:00:27	
	DownstreamProject/	-	2020-10-12 07:00:09	
	IntegrationTests/	-	2020-10-12 07:00:09	
	UnitTests/	-	2020-10-12 07:00:09	
	CMakeLists.txt	240 bytes	2020-10-12 07:00:09	-

Code Browsing



EIC code displayed by LXR

master/fun4all_macros/detectors/fsPHENIX/Fun4All_G4_fsPHENIX.C

```
0001 #ifndef MACRO_FUN4ALLG4FSPHENIX_C
0002 #define MACRO_FUN4ALLG4FSPHENIX_C
0003
0004 #include "GlobalVariables.C"
0005
0006 #include "DisplayOn.C"
0007 #include "G4Setup_fsPHENIX.C"
0008 #include "G4_Bbc.C"
0009 #include "G4_CaloTrigger.C"
0010 #include "G4_DSTReader_fsPHENIX.C"
0011 #include "G4_FwdJets.C"
0012 #include "G4_Global.C"
0013 #include "G4_Input.C"
0014 #include "G4_Jets.C"
0015 #include "G4_Production.C"
0016
0017 #include <fun4all/Fun4AllDstOutputManager.h>
0018 #include <fun4all/Fun4AllOutputManager.h>
0019 #include <fun4all/Fun4AllServer.h>
```

Added feature, links to internal included files:

- ◆ Unique file: link to the file
- ◆ Multiple occurrences, check the path:
 - ✓ Best path matching found, link to the match
 - ✓ Otherwise, link to the filename search

Links to identifier search

```
0129     (Input::UPSILON)
0130     {
0131     INPUTGENERATOR::VectorMesonGenerator->add_decay_particle
0132     INPUTGENERATOR::VectorMesonGenerator->set_rapidity_range
0133     INPUTGENERATOR::VectorMesonGenerator->set_pt_range(0.,
0134     // Y species - select only one, last one wins
0135     INPUTGENERATOR::VectorMesonGenerator->set_upsilon_1s();
0136     }
0137     // particle gun
0138     Input::GUN)
0139     INPUTGENERATOR::Gun->AddParticle("pi-", 0, 1, 0);
0140     INPUTGENERATOR::Gun->set_vtx(0, 0, 0);
0141 }
```


Identifier Search

Identifier search

Type the full name of an identifier to look for (a function name, variable name, class name, etc.). Matches are case-sensitive. Check "Definitions only" to find only definitions. As a convenience, extra +999 line numbers denote case-insensitive matches.

**Identifiers:
variable, class, function**

Identifier: Definitions only

Definitions for DISPLACED_VERTEX

Type	Member of	File	Line
class, struct, or union member		/fun4all_macros/common/G4_Tracking_EIC.C	35
class, struct, or union member	G4TRACKING	/fun4all_macros/detectors/EICDetector/Fun4All_G4_EICDetector.C	219
class, struct, or union member	G4TRACKING	/fun4all_macros/macros/g4simulations/Fun4All_G4_EICDetector.C	146
class, struct, or union member		/fun4all_macros/macros/g4simulations/G4_Tracking_EIC.C	34
local variable		/g4lblvtx/macros/Fun4All_G4_FastMom.C	55
local variable		/g4lblvtx/macros/Fun4All_G4_FastMom_GEM.C	56
local variable		/g4lblvtx/macros/Fun4All_G4_FastMom_GEM_RICH.C	57

7 declarations in 7 files.

References to DISPLACED_VERTEX

File	Line
/fun4all_macros/common/G4_Tracking_EIC.C	35 59
/fun4all_macros/detectors/EICDetector/Fun4All_G4_EICDetector.C	219
/fun4all_macros/macros/g4simulations/Fun4All_G4_EICDetector.C	146
/fun4all_macros/macros/g4simulations/G4_Tracking_EIC.C	34 59
/g4lblvtx/macros/Fun4All_G4_FastMom.C	55 232
/g4lblvtx/macros/Fun4All_G4_FastMom_GEM.C	56 260
/g4lblvtx/macros/Fun4All_G4_FastMom_GEM_RICH.C	57 264

12 references in 7 files.

It displays **definitions** and **References**

Text Search

Powered by [Glimpse](#). (Tips for search syntax.)

126 occurrences found.

Powered by [Glimpse](#)

Results for `GetEvent`

File	Line	Code
EicRoot/PndTools/generalTools/PndEventCounterTask.h	22	<code>Int_t GetEventCount() const {return fEvtCounter;}</code>
EicRoot/base/FairEventHeader.h	29	<code>Double_t GetEventTime() {return fEventTime;}</code>
EicRoot/base/FairMCApplication.cxx	261	<code>//FairMCEventHeader* header = gen->GetEvent();</code>
EicRoot/base/FairMCEventHeader.h	47	<code>UInt_t GetEventID() const { return fEventId; } /// event</code>
EicRoot/base/FairMCPoint.h	42	<code>UInt_t GetEventID() const { return fEventId; }</code>
EicRoot/base/FairPrimaryGenerator.cxx	165	<code>if (-1 == fEvent->GetEventID()) {</code>
EicRoot/base/FairPrimaryGenerator.cxx	170	<code>fLogger->Info(MESSAGE_ORIGIN, "FairPrimaryGenerator: (Event %i</code>
EicRoot/base/FairPrimaryGenerator.h	122	<code>FairMCEventHeader* GetEvent() { return fEvent; };</code>
EicRoot/base/FairRootManager.cxx	1076	<code>fEvtHeader->SetEventTime(GetEventTime());</code>
EicRoot/base/FairRootManager.cxx	1097	<code>fEvtHeader->SetEventTime(GetEventTime());</code>
EicRoot/base/FairRootManager.cxx	1144	<code>fEvtHeader->SetEventTime(GetEventTime());</code>
EicRoot/base/FairRootManager.cxx	1155	<code>fEvtHeader->SetEventTime(GetEventTime());</code>

Version Comparison



EIC code displayed by LXR

master/fun4all_macros/detectors/EICDetector/Fun4All_G4_EICDetector.C

Diff markup

between /fun4all_macros/detectors/EICDetector/Fun4All_G4_EICDetector.C (Version: master)
and /fun4all_macros/detectors/EICDetector/Fun4All_G4_EICDetector.C (Version: test)

```
0001 #ifndef MACRO_FUN4ALLG4EICDETECTOR_C      0001 #ifndef MACRO_FUN4ALLG4EICDETECTOR_C
0002 #define MACRO_FUN4ALLG4EICDETECTOR_C      0002 #define MACRO_FUN4ALLG4EICDETECTOR_C
0003                                           0003
0004 #include "GlobalVariables.C"              0004 #include "GlobalVariables.C"
0005                                           0005
0006 #include "DisplayOn.C"                    0006 #include "DisplayOn.C"
0007 #include "G4Setup_EICDetector.C"         0007 #include "G4Setup_EICDetector.C"
0008 #include "G4_Bbc.C"                      0008 #include "G4_Bbc.C"
0009 #include "G4_CaloTrigger.C"              0009 #include "G4_CaloTrigger.C"
0010 #include "G4_DSTReader_EICDetector.C"    0010 #include "G4_DSTReader_EICDetector.C"
0011 #include "G4_FwdJets.C"                  0011 #include "G4_FwdJets.C"
0012 #include "G4_Global.C"                   0012 #include "G4_Global.C"
0013 #include "G4_HIJetReco.C"               0013 #include "G4_HIJetReco.C"
0014 #include "G4_Input.C"                   0014 #include "G4_Input.C"
0015 #include "G4_Jets.C"                     0015 #include "G4_Jets.C"
0016 #include "G4_Production.C"               0016 #include "G4_Production.C"
0017                                           0017
0018 #include <TROOT.h>                         0018 #include <fun4all/Fun4AllDstOutputManager.h>
0019 #include <fun4all/Fun4AllDstOutputManager.h> 0019 #include <fun4all/Fun4AllOutputManager.h>
0020 #include <fun4all/Fun4AllOutputManager.h> 0020 #include <fun4all/Fun4AllServer.h>
0021 #include <fun4all/Fun4AllServer.h>         0021
0022                                           0022
0023 #include <phool/recoConsts.h>              0023 #include <phool/recoConsts.h>
0024                                           0024
```

File Types in Glimpse

Glimpseindex **indexes only text files**. It attempts to identify non-text files such as binary files, compressed files, uuencoded files, postscript files, binhex files, etc.

The **binary file** check is done by **sampling** files:

- ❖ For example, would be identified as **binary** if **>10% lines longer than 127 characters, which is not documented well**.

It misidentified some source code text files.

Fortunately, there is an **option -E** to skip the file type check.

Yes, the option fixed the problem, and did not make the indexing much longer (after having removed the .git/ subdirs and .root files).

ctags

Identifier/Reference indexing is done through ctags.

The current installed and used by LXR is Exuberant Ctags:

- ❖ Support >40 languages
- ❖ But the maintenance has stopped since 2009

Universtal Ctags forked this ctags, and continues the development, with many improvements/features:

- ❖ Support >100 languages
- ❖ Improved C/C++/Python parser

We are going to replace with the Universal Ctags.

Have checked that it is compatible with LXR.

Plans for EIC LXR

- ❖ Update ctags with Universal Ctags and explore its new features
(Some effort needed)
- ❖ Include other external packages such as ROOT and CLHEP? (but difficulty with version choosing)
So relevant reference links would be available.
- ❖ Fix small problems per user requests
E.g. caused by erroneous or odd syntax