



gAn tutorial

Germano & Ruggero

AEgIS Analysis Framework

CERN - 21/09/2016

Premise

- the AEGIS **raw data** are written by the DAQ in a ROOT file
- for “analysis” we mean accessing the **raw data** and producing numbers and distributions
- at the moment each users doing analyses is using his own program (mainly root macros)

gAn objective

- to have a common Framework for all the AEGIS analyses
- gAn is working but it is still in beta version**

gAn basics

- the gAn framework it is based on ROOT and has been developed in C++
- the gAn reads the raw data and produces numbers and distributions
- it is intended to be used by everybody (even with little or no knowledge of C++)
- it can be expanded and enriched by users with some knowledge of C++ (*developers*) and the additions can be easily included in the framework for all the other users

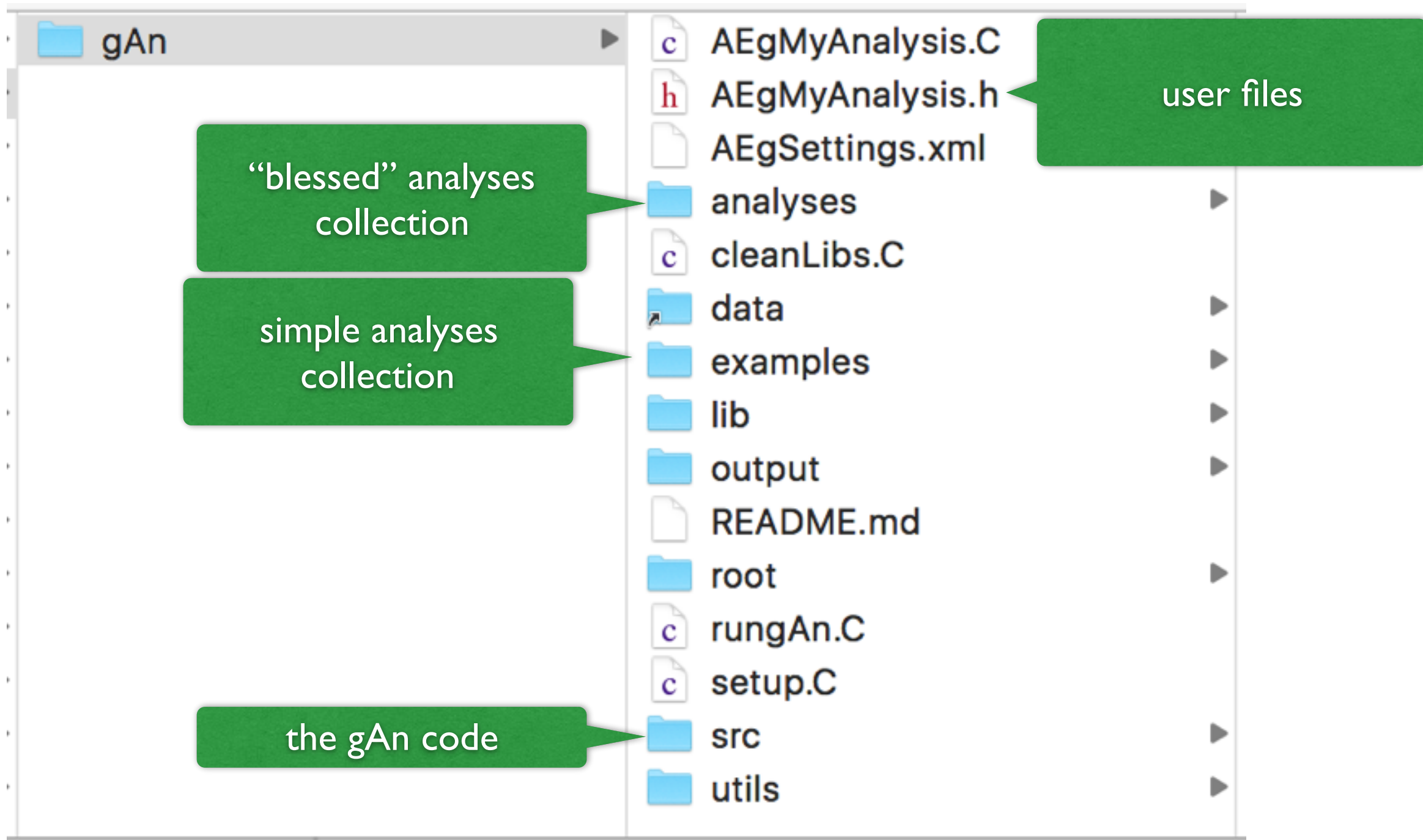
Tutorial summary

- framework general description
- code management
 - > GitLab repository (git)
 - > Access levels
- general use
 - > documentation
 - > how to get the gAn and how to get the data
 - > how to run it (live tutorial)
 - > the AEgSettings.xml (live tutorial)
 - > how to run an available analysis (live tutorial)
 - > how to write your own analysis (live tutorial)
- Q&A



framework general description

the “directory” structure





framework general description

the User Code is handled by the `AEgMyAnalysis` class (inherited from `AEgAnalysisBase`)

`AEgMyAnalysis.C` and `AEgMyAnalysis.h` files are not part of the repository - they are copied from `utils/AEgTemplate.C` and `AEgTemplate.h` at the setup stage

In this way every user can develop his own analysis, without changing the code in the repository.

Users can then exchange `AEgTemplate.C` and `AEgTemplate.h` so that every other user can test his analysis

the user run gAn via the `rungAn` macro [`.x rungAn.C(run)` or `.x rungAn("user file list.txt")`]

actually `rungAn.C` loads the class libraries and calls `execgAn.C` that in its turn load the `AEgMyAnalysis` class and launch the analysis [this workaround is needed by ROOT6]

`AEgMyAnalysis`

`AEgRun` (1 object per run)

`AEgIO` (deals the interface with the root(s) file)

RAW DATA ROOT FILE

`AEgImage` class
`AEgCCD` and `AEgMimito`
are derived classes

`AEgTrace` class
`AEgPMT` and `AEgFarCup`
are derived classes

`AEgScaler` class
`AEgScint`
is a derived class

`RunInfo` struct
(number, time,
class, type)

the “main” program

```
void execgAn(Int_t run_start, Int_t run_end) {
    // Initialize settings system
    AEgSettingsManager::Instance().LoadConfigFile("AEgSettings.xml");

    // Create user code
    AEgMyAnalysis *User = new AEgMyAnalysis();

    // Open output file
    User->CreateOutputFile(run_start, run_end);

    // Cycle on the list of runs
    for (int run=run_start; run<=run_end; ++run) {
        // Prepare settings for this run number
        AEgSettingsManager::Instance().SetRunNumber(run);

        // Open the run
        AEgRun *Run = new AEgRun(run);

        // Check if it is valid
        if(!Run->GetIsRun())
            continue;

        // Process the analysis
        bool Ret = User->AnalyzeRun(Run);

        // If run was marked valid, save the data
        if (Ret)
            User->PushAnalyzedData(Run);

        // Check if in batch mode
        if (gBatchMode) {
            // Delete run
            delete Run;
        }
    }
}
```

it reads the
user's settings

it creates the
user's “analysis”

it creates the “AEgRun” object
(raw data decoding etc.)

the “analysis” is performed

it save the data

go to the next “run”



framework general description

the AEgSettings.xml file

```
1  <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
2
3  <!-- General settings of gAn -->
4  <general>
5      <verbosity    v="1" />
6      <batch_mode   v="0" />
7  </general>
8
9
10 <!-- Paths where to access the files -->
11 <path>
12     <enviro_data  v="enviro/" />
13     <run_data     v="data/" />
14     <out_data     v="output/" />
15
16 <!--
17     <run_data v="data/run_2012/" from="0" to="9999" />
18     <run_data v="data/run_2013/" from="0" to="9999" />
19     <run_data v="data/run_2014/" from="0" to="9999" />
20     <run_data v="data/run_2015/" from="0" to="9999" />
21     <run_data v="data/run_2016/" from="0" to="9999" />
22 -->
23 </path>
24
25
26 <!-- Output configuration -->
27 <output>
28     <file          v="gAnOut" />
29     <mode          v="recreate" />
30     <tree_name     v="out" />
31     <tree_desc     v="Analyzed gAn data" />
32 </output>
33
34
35 <!-- Analysis base configuration -->
36 <analysis_base>
37     <enable_mimito v="1" />
38     <enable_scint  v="1" />
39     <enable_mcp    v="1" />
40     <enable_hdmcp  v="1" />
41     <enable_pmt    v="0" />
42     <enable_farcup v="1" />
43 </analysis_base>
```




gAn-web

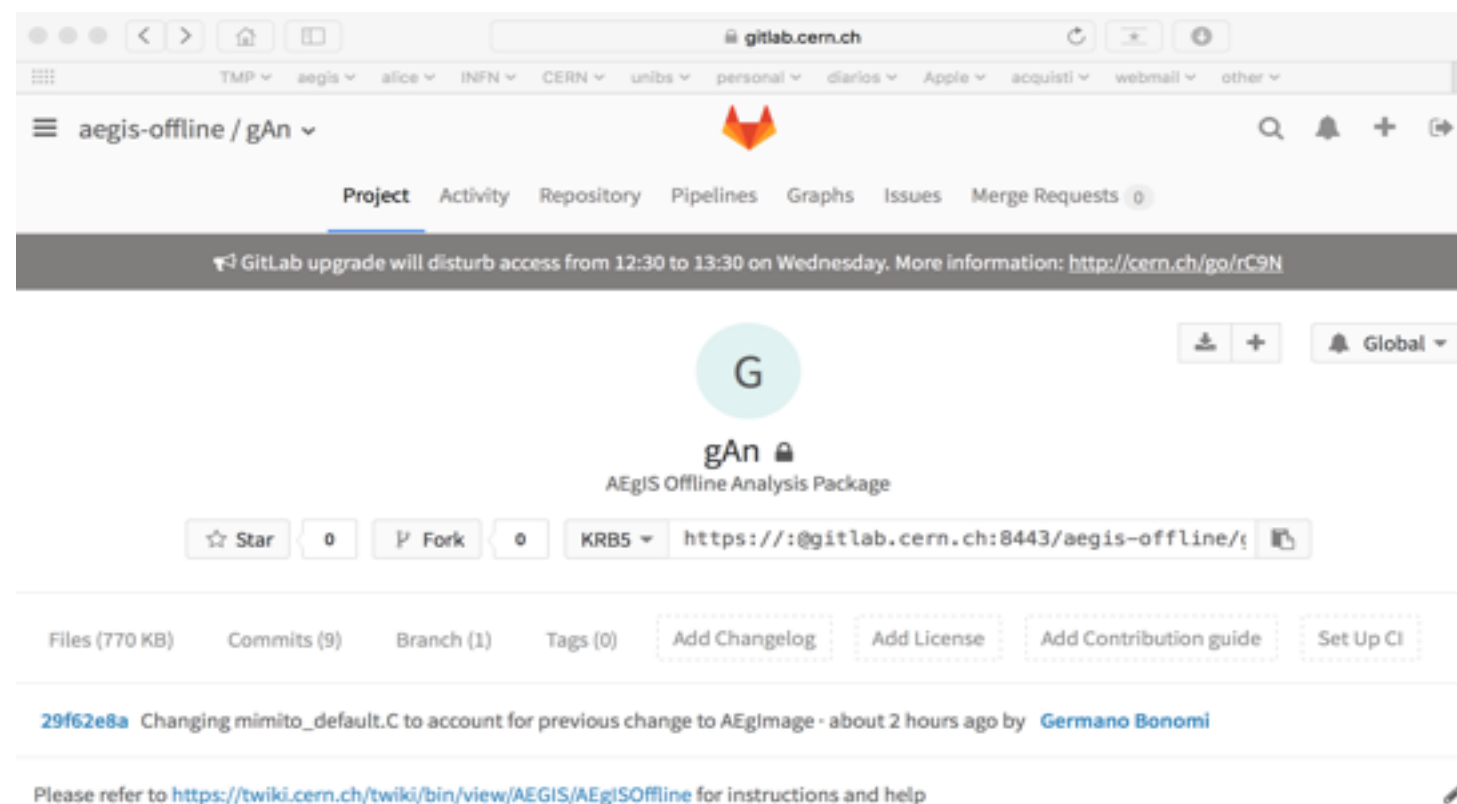
framework general description

There is a “side-project” called gAn-web that has the goal to develop a web interface to gAn. It would be a “server-side” program that would run “default” analyses and produce “standard” plots. For such outputs there would be no need to installation and configuration, just typing a run number and push a button

The screenshot shows a web browser window with the address bar displaying 'aegisgateway.cern.ch'. The page has a light blue background and features the large text 'A E \bar{g} I S' in the center. Below this, it says 'Choose a run. Last existing run: 47002, from: 21/09/2016'. There is a text input field with the placeholder text 'example: 30000; 30001; 30003' and a green 'Send' button. To the right of the input field, it says 'You are using: gAn-dev'. At the bottom left, there is a button labeled 'Add runs by range' and a blue button labeled 'Edit Configuration File'.



code management



The **gAn** is hosted in the CERN GitLab [aegis-offline GitLab group]

- The project **is** accessible via 3 levels of permissions
 - **users** -> able to install and use the package (writing his own analysis)
 - **developers** -> users that can create new branches and develop code [but not able to commit and push to protected branches]
 - **masters** -> managers of the repository

some **git** knowledge is necessary only for developers but not for users



general use

documentation

The **gAn** manual page is available from our TWiki page:
<https://twiki.cern.ch/twiki/bin/view/AEGIS/AEgISOffline>

The screenshot shows a web browser window displaying a TWiki page. The browser's address bar shows 'twiki.cern.ch'. The page has a navigation bar with various site links like 'TMP', 'aegis', 'alice', 'INFN', 'CERN', 'unibs', 'personal', 'diarios', 'Apple', 'acquisti', 'webmail', and 'other'. A search bar is present with a 'Jump' field and a 'Search' button. The page content is titled 'Welcome to the gAn - AEgIS Offline Analysis Framework'. It includes a list of links: Introduction, Prerequisites, Getting the code, Setup, Running the code, gAn documentation, gAn development, gAn coding conventions, and Standard analysis. The 'Standard analysis' link has sub-links: Run general information and Run selection. The 'Introduction' section is expanded, showing text about downloading and using the framework, and a contact email: aegis-offline-masters@cernSPAMNOT.ch. The 'Prerequisites' section is also visible.

Work in progress ...



general use

documentation

The **gAn** class description is available here:

<http://aegis.web.cern.ch/aegis/aegis-offline/gAn/html/index.html>

The screenshot shows the web interface of the gAn: AEgIS Analysis Framework documentation. The browser address bar shows 'aegis.web.cern.ch'. The page has a navigation menu with tabs for 'Main Page', 'Namespaces', 'Classes', and 'Files'. The 'Classes' tab is active, and a sub-menu shows 'Class List', 'Class Index', 'Class Hierarchy', and 'Class Members'. The 'Class List' section is displayed, showing a list of classes and their brief descriptions. The list includes:

- tinyxml2
- AEgAnalysisBase
- AEgBaseSetting
- AEgConfigFile
- AEgFarCup
- AEgImage
- AEgImageFilter
- AEgImageReducer
- AEgIO: AEgIO Class - Interface class between the user and the AEgIS raw data root file
- AEgMCP
- AEgMimito: AEgMimito Class - Derived from AEgImage class
- AEgMyAnalysis: Class that contains the user code
- AEgPMT: AEgPMT Class - Derived from AEgTrace class
- AEgRun: AEgRun Class - Interface class between the user and AEgIO
- AEgScaler: AEgImage Class - It defines the AEgIS template for scaler data (such as SIS data)
- AEgScint: AEgScint Class - It derives from AEgScaler

Work in progress ...



general use

first steps

```
git clone https://gitlab.cern.ch/aegis-offline/gAn.git
cd gAn
root setup.C
```

how to get the raw data [the data need to be copied in the “data” sub-folder]

The **raw data** are available from the web:

https://aegisgateway.cern.ch:8443/daq/root_files/?C=M;O=D

The **enviro data** are available from the web:

https://aegisgateway.cern.ch:8443/daq/root_files/environmental_monitor/

In alternative they can be copied via *scp* or *sftp* from one of the linux machines in the control room (i.e. *aegislx01*, *aegislx04*, *aegisc101*) from the /data directory

... now you are ready to run the  ...