

# The EICUG Software Website - an update

Maxim Potekhin  
(BNL, NPPS)  
09/02/2020

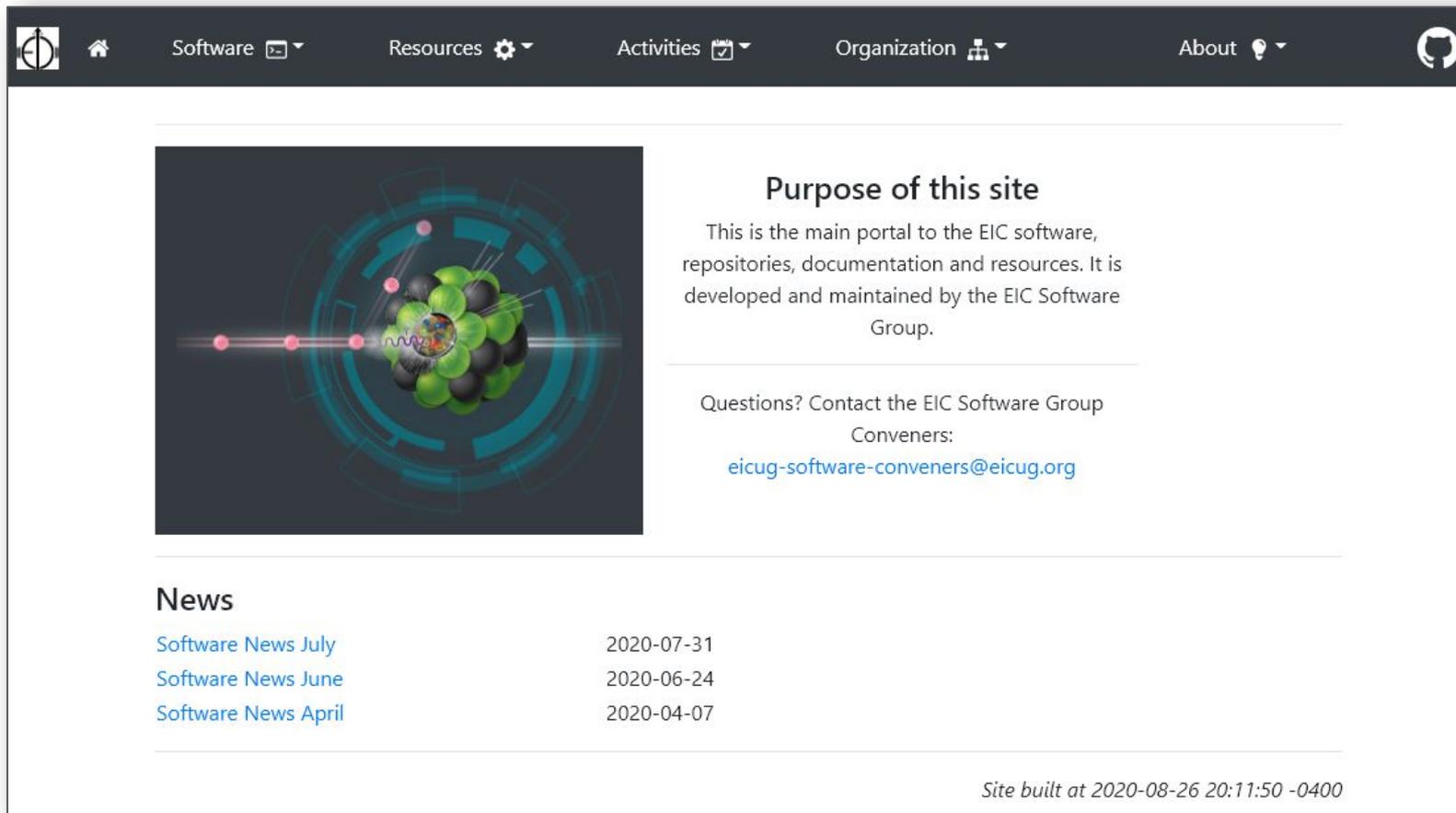
# Why are we here?

- For many organizations their website is an important tool and an asset.
- More so for knowledge-intensive fields like ours.
- Let's take stock of where we are with the website, decide on what can be improved or changed, and define items for future development.
- The content is a separate topic, these slides are mostly “how-to”.

# Overview

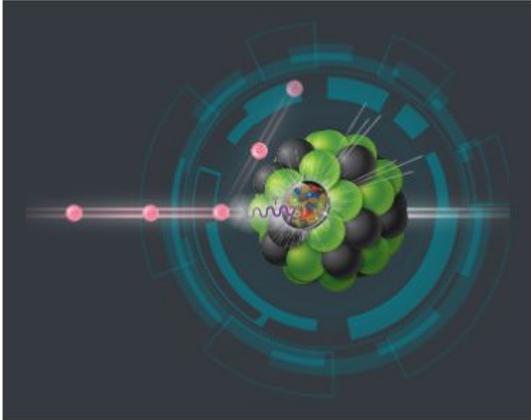
- The website has been online for ~6 months <https://eic.github.io/>
- See presentation from April 2020 for intro and general idea of the site
  - [https://docs.google.com/presentation/d/1WCVIVNZldHm8exHJBhILLngr9yEXHUP1w\\_BdwB4F41s/edit?usp=sharing](https://docs.google.com/presentation/d/1WCVIVNZldHm8exHJBhILLngr9yEXHUP1w_BdwB4F41s/edit?usp=sharing)
- Healthy level of contributions to the site from the team members
  - A proof that the system is reasonably user friendly
- Changes to the layout can be done quite quickly
  - Not a problem to create a new skin, menu, color scheme, layout etc
- GitHub based
  - Free hosting with service level we are happy with for now
  - Pull requests make collaboration easy
- How are things looking going forward? What have we learned?

# Updated layout/menus - recent changes



The screenshot displays the website's navigation menu at the top, including links for Software, Resources, Activities, Organization, and About, along with a GitHub icon. The main content area features a large image of a molecular structure, a section titled "Purpose of this site" explaining the portal's role, contact information for the EIC Software Group, and a "News" section with links to recent updates.

**Software** **Resources** **Activities** **Organization** **About**



## Purpose of this site

This is the main portal to the EIC software, repositories, documentation and resources. It is developed and maintained by the EIC Software Group.

Questions? Contact the EIC Software Group  
Conveners:  
[eicug-software-conveners@eicug.org](mailto:eicug-software-conveners@eicug.org)

## News

<a href="#">Software News July</a>	2020-07-31
<a href="#">Software News June</a>	2020-06-24
<a href="#">Software News April</a>	2020-04-07

*Site built at 2020-08-26 20:11:50 -0400*

# Updated layout/menus, cont'd

The image displays a website interface with a dark header and a light main content area. The header contains navigation links: Software, Resources, Activities, and Organization. A dropdown menu is open under 'Software', listing categories like Frameworks and Tools, Singularity Applications, Monte Carlo Event Generators, and Infrastructure. Another dropdown menu is open under 'Organization', listing items like Software Working Group, Communication, Meetings and Events, Our YouTube Channel, and Get Involved. The main content area features a section titled 'Purpose of this site' with a paragraph of text and a contact email address. A footer note indicates the site was built on 2020-08-26.

Organization 

Software  Resources  Activities  Organization 

Frameworks and Tools

- ESCalate
- Fun4All
- EIC-smear

Singularity Applications

- Escalate
- EIC-smear and Generators
- Fun4All in Singularity 

Monte Carlo Event Generators

- PYTHIA6
- PEPSI

Other Software

- BeAST Magnetic Field

Infrastructure

- Doxygen 
- GitHub organization for the EIC User Group 

**Purpose of this site**

This is the main portal to the EIC software repositories, documentation and resources developed and maintained by the EIC Software Group.

Questions? Contact the EIC Software Group Conveners:  
[eicug-software-conveners@eicug.org](mailto:eicug-software-conveners@eicug.org)

2020-07-31  
2020-06-24  
2020-04-07

Site built at 2020-08-26 20:11:50 -0400

# The platform

- Static site = security + performance.
- Template-based, hence separation of content and layout
- Jekyll is one of a few available static website generators.
- Markdown/Jekyll/Liquid/Bootstrap + GitHub.
- Jekyll does allow for a fairly liberal mix of HTML and MD for considerable flexibility
  - ...with a few caveats
- YAML, CSV and JSON files containing configuration and data are transparently parsed - creating data structures in an intuitive way
  - Example: migration of data from Postgres requires one line of sql and nothing else
- Manipulation of structured data is perhaps the most important feature.

# Structured data

- Structured data + code written in Liquid language = **procedural generation** of static content on the website.
- Similar to what can be done with a DB (admittedly more awkward to code), but without the actual DB, just structured data files (YAML etc)
  - They key is to do an equivalent of a “JOIN” which we use in RDBMS
- In many cases this translates into
  - Efficiency of development
  - Ease of maintenance
  - **Referential integrity** of content (i.e. data is not duplicated on the pages of the site but is created once and then used by reference)
  - Efficient content reuse

# Versatility

- Since YAML (or JSON) organically include the dictionary/map concept there is a lot of versatility of how the data can be designed and used.
- Gallery content, contact lists, software test results, validation, references to documentation and external links etc
  - Example - don't put a hard link to a local image file on a page, add it to the gallery file and refer to it by a mnemonic name/shortcut
  - Automation of cumbersome external links in various places on the site, add it to the links file and refer to it by a mnemonic name/shortcut
- Keyword queries work nicely if keywords are added to the data.
- To make all of this possible, the site must be instrumented with macros a.k.a. Jekyll include files (look at the `_include` folder for example).

# Data files in the repository

🔍 master ▾ [eic.github.io / \\_data /](https://eic.github.io/_data/) Go to file Add file ▾

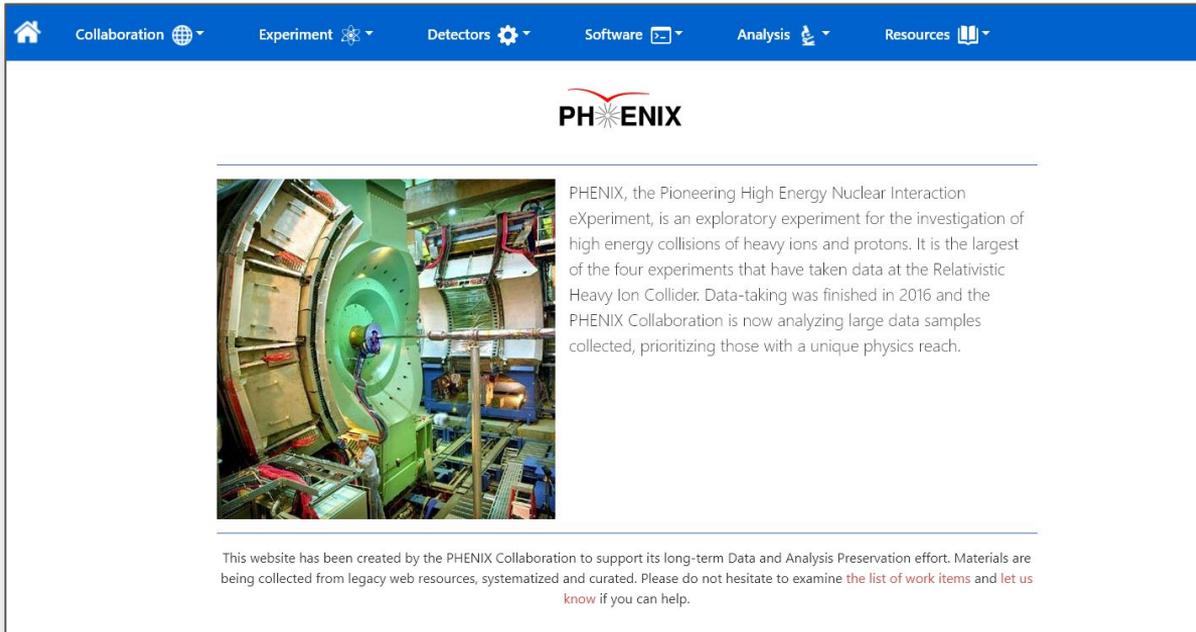
 **buddhasystem** finished stubbing out "organization", and the general layout suggeste... ... ✓ a6bc107 6 days ago 🕒 History

..

 README.md	Added the stub for the software registry, removed reference to the so...	231 B	5 months ago
 gallery.yml	Added the updated image handling macro to a few pages	264 B	14 days ago
 links.yml	finished stubbing out "organization", and the general layout suggeste...	1.78 KB	6 days ago
 menus.yml	finished stubbing out "organization", and the general layout suggeste...	3.91 KB	6 days ago
 people.yml	stubbed out new content for the "people" dropdown, formerly "teams"	1.79 KB	4 months ago
 software.yml	Add all fun4all tutorials	775 B	3 months ago

# Experience

- The new PHENIX website developed in parallel with the EICUG Software
  - Similar code and logic
- A lot more material than on our site (and still plenty to migrate)



PHENIX

PHENIX, the Pioneering High Energy Nuclear Interaction eXperiment, is an exploratory experiment for the investigation of high energy collisions of heavy ions and protons. It is the largest of the four experiments that have taken data at the Relativistic Heavy Ion Collider. Data-taking was finished in 2016 and the PHENIX Collaboration is now analyzing large data samples collected, prioritizing those with a unique physics reach.

This website has been created by the PHENIX Collaboration to support its long-term Data and Analysis Preservation effort. Materials are being collected from legacy web resources, systematized and curated. Please do not hesitate to examine [the list of work items](#) and [let us know](#) if you can help.

# What works and what doesn't

- What works
  - Extensive use on YAML/CSV to organize all sorts of information
  - Automated menu/navigation system
  - Customized CSS
  - Keeping site wide definitions (of any kind) in *\_config.yaml*
  - Templates for specific page types
  - **Macros** (includes) - for many purposes, also helps to leverage HTML when MD falls short, and to achieve consistent styling (which is important)
- What doesn't work
  - Using the site as a document and media server
    - The 'GitHub Pages' caps: 1GB repo, 1GB site content and 100GB/month traffic
    - Fat repos become cumbersome to operate, fills up much faster than you expect
    - Indexing capabilities are completely inferior compared to Zenodo etc
  - Hard links to internal and external pages, document references, images - things do change
  - Hardcoded menus
  - Unnecessary use of HTML (cf MD is not rendered within HTML blocks)

# Automated generation of content/links - by tags

## Electromagnetic Calorimeter

---

### Write-ups

- DOI [10.5281/zenodo.3833205](https://doi.org/10.5281/zenodo.3833205) PHENIX Electromagnetic Calorimeter (EMCal) – Detector Basics (G.David)
- DOI [10.5281/zenodo.3893972](https://doi.org/10.5281/zenodo.3893972) Explanation of PHENIX triggers (A.Bazilevsky)

### Theses

- DOI [10.5281/zenodo.3885856](https://doi.org/10.5281/zenodo.3885856) The Quark Gluon Plasma probed by Low Momentum Direct Photons in Au+Au Collisions at  $\sqrt{s_{NN}}=62.4\text{GeV}$  and  $\sqrt{s_{NN}}=39\text{ GeV}$  beam energies (Vladimir Khachatryan)
- DOI [10.5281/zenodo.3885870](https://doi.org/10.5281/zenodo.3885870) Inclusive jet production in proton-proton and copper-gold collisions at  $\sqrt{s_{NN}} = 200\text{ GeV}$  (Arbin Timilsina)

### Publications

- [PHENIX Calorimeter](#) (NIM A 499, 2003, doi.org/10.1016/S0168-9002(02)01954-X)
- [High Energy Beam Test of the PHENIX Lead-Scintillator EM Calorimeter](#) High Energy Beam Test of the PHENIX Lead-Scintillator EM Calorimeter

### Presentations

- DOI [10.5281/zenodo.4007113](https://doi.org/10.5281/zenodo.4007113) PHENIX Focus: Electromagnetic Calorimeter (Gabor David)

### Variables and Accessors under PHCentralTrack Node (used for charged particle analyses)

Type	Name	Description
float	get_pemcx	x-component of the projection of the cgl track onto the EMC (cm)
float	get_pemcy	y-component of the projection of the cgl track onto the EMC (cm)

# Helper macros

- <https://eic.github.io/about/howto.html#helper-macros> paraphrased syntax:
- *include pagelink.md name='my\_page'*
- *include image.md name='my\_pic'*
  
- Please check the examples in the link
- The toolkit will grow

## Helper macros

### Links

Frequently used **external** links are located in the following registry: [\\_data/links.yml](#).

They can be easily inserted in pages in a consistent manner by their mnemonic name using a macro as described below. This simplifies handling user-unfriendly links and ensures they remain the same across the site and makes it easy to point to a different resource when necessary. For example, this piece of code

```
{% include navigation/findlink.md name='Jekyll' %}
```

will result in the link: [Jekyll](#). If you would like the displayed link name to be different, the optional "tag" argument for the macro will do that. Example:

```
{% include navigation/findlink.md name='github_site' tag='repository' %}
```

will result in the following link: [repository](#). Links will automatically open in a new tab.

There is a similar macro for use with links to **internal** pages on the site. Example:

```
{% include navigation/pagelink.md folder=site.about name='howto' tag='How-to' page' %}
```

will result in: ["How-to" page](#) (this happens to point to the page you are reading now). Primary advantage of this macro is that it allows renaming the source file - while keeping the same "name" attribute in the Front Matter section the links will still be correct.

### Images

Users and developers have complete freedom in how they incorporate images into pages on this site. In many cases handling images will be facilitated by adding an image to the following registry: [\\_data/gallery.yml](#) and then using a simple macro to refer to the image by its mnemonic name and automatically generate the correct link, as illustrated in the following example:

```
{% include images/image.md name='news_banner' width='400' %}
```

...will produce



# Serving media (including PDF)

- Best solution currently available: Zenodo
- Until it is adopted, the following can be used
  - A separate GitHub repository for documents like PDF files (has preview)
    - Good experience in PHENIX with that
  - Dropbox
- Need to keep local content on the website to a bare minimum

# Documenting software

- READMEs on GitHub vs the site content
- My 2 cents - READMEs were never meant to be textbooks on complex software subjects
  - Meant to contain basic info e.g. intro, “installation from sources”, license etc
- Good projects always keep separate documentation pages elsewhere
- In case of the EICUG software it appears to be helpful to cross-reference software and tools in one place
- Relying on the GitHub READMEs leads to fragmentation of documentation

# Summary

- The website is in a good shape and is very adaptable
- Let's keep media (including PDF) separately so that the site repository is lean
- Making contributions (via PR) is easy
- Please make suggestions of what structured data needs to be added (list of validations etc)