IEEE SA Open

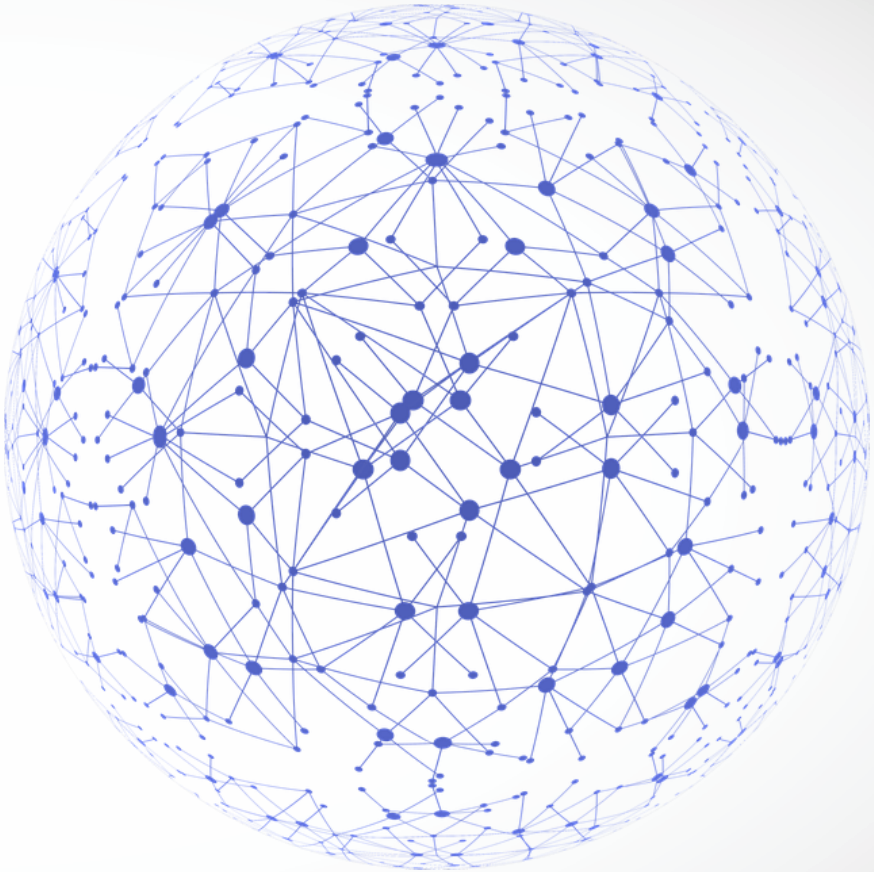
Tutorial for 802 Plenary

March 2025

Robby Robson
Sarah Womer
Kent Lusted

Agenda

- The Speakers
- Intro to Open Source
- Open Source in Standards
- SA Open
- Policies and Procedures
- Discussion



The Speakers

Robby Robson

Affiliated with Corvallis Group, LLC & Apple

- Co-founder / former CEO of Eduworks
  Led open source development teams
- First chair of OSCom
  Led team that developed first OSCom P&P
- 2025 member SASB / ProCom Chair
- 2025 chair Industry Engagement Committee Workforce Working Group

Sarah Womer

Affiliated with Plessas Experts Network Inc.

- Senior Analyst Plessas Experts Network Inc.
- 20+ Years in Open Source Intelligence and Online Research
- OSCom member and IEEE SA Open volunteer

Kent Lusted

Affiliated with Synopsys

- 15+ year active contributor and member of the IEEE 802.3 leadership team
- IEEE 802.3 P802.3dj Task Force Electrical track Chair
- IEEE 802.3 COM Open Source Ad Hoc Chair

Intro to Open Source

What is “Open Source” Anyway?

How is Open Source Developed?

What is “Open Source” Anyway?

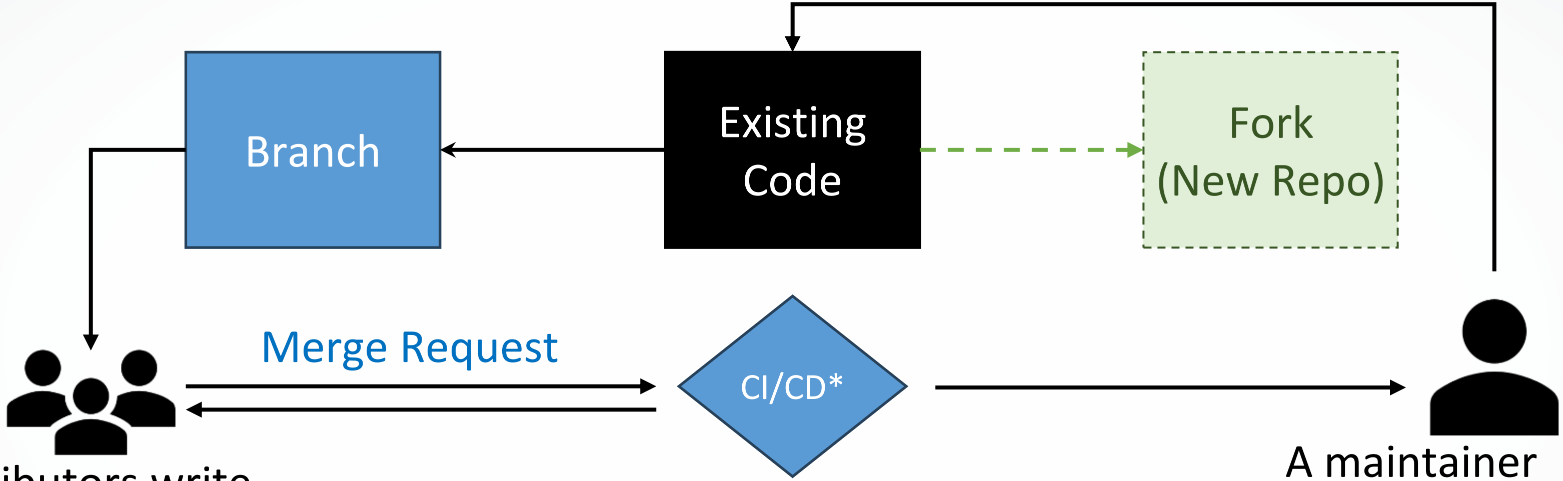
https://opensource.org/osd

- Source code is available for anyone to see, modify, and distribute
- Governed by an open source license (and contributor license agreement, or CLA)

But there is more …

- Variations on licenses and CLAs
  - Licenses defined by the Open Source community in various ways
    - “Copyleft” or “viral” licenses (e.g. Gnu Public License)
    - “Permissive” or “business friendly” licenses (e.g. MIT, BSD, or Apache)
    - May (or may not) address patents
- Variations on governance that determines how open source is developed and updated
  - Large communities governed as meritocracies or democracies
  - Tightly controlled by a single organization (or a benevolent dictator)
  - Governed by consensus within a Working Group
  - See https://www.theopensourceway.org/

How is Open Source Developed?



Branch

Existing Code

Fork (New Repo)

Merge Request

CI/CD*

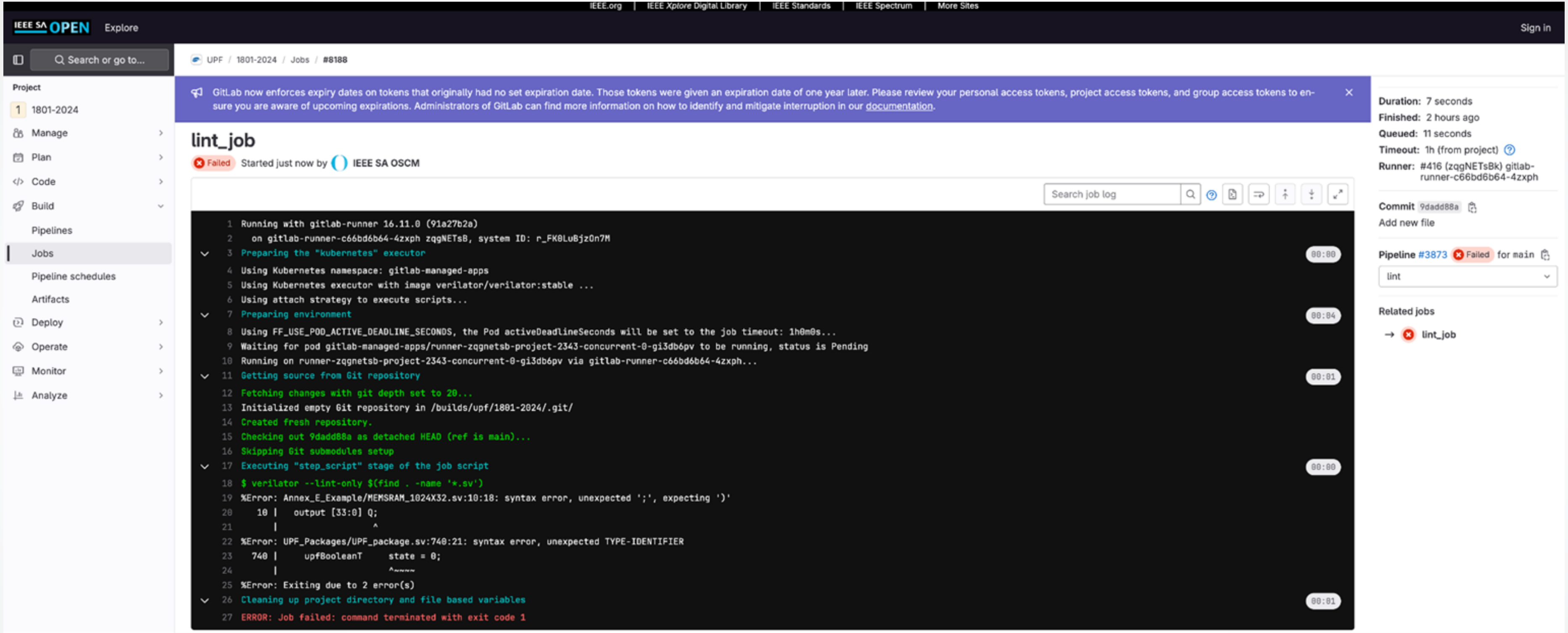
Contributors write new code or modify existing code

Automated tests run …

*Continuous Integration / Continuous Delivery

A maintainer merges the new or modified code into the existing branch

Orchestrated by a *maintainer* who is responsible for the overall health of the project



GitLab now enforces expiry dates on tokens that originally had no set expiration date. Those tokens were given an expiration date of one year later. Please review your personal access tokens, project access tokens, and group access tokens to ensure you are aware of upcoming expirations. Administrators of GitLab can find more information on how to identify and mitigate interruption in our documentation.

lint_job

Failed Started just now by IEEE SA OSCM

```
Running with gitlab-runner 16.11.0 (91a27b2a)
  on gitlab-runner-c66bd6b64-4zxph zqgNETsB, system ID: r_FK0LuBjzOn7M
Preparing the "kubernetes" executor                                    00:00
Using Kubernetes namespace: gitlab-managed-apps
Using Kubernetes executor with image verilator/verilator:stable ...
Using attach strategy to execute scripts...
Preparing environment                                                  00:04
Using FF_USE_POD_ACTIVE_DEADLINE_SECONDS, the Pod activeDeadlineSeconds will be set to the job timeout: 1h0m0s...
Waiting for pod gitlab-managed-apps/runner-zqgnetsb-project-2343-concurrent-0-gi3db6pv to be running, status is Pending
Running on runner-zqgnetsb-project-2343-concurrent-0-gi3db6pv via gitlab-runner-c66bd6b64-4zxph...
Getting source from Git repository                                     00:01
Fetching changes with git depth set to 20...
Initialized empty Git repository in /builds/upf/1801-2024/.git/
Created fresh repository.
Checking out 9dadd88a as detached HEAD (ref is main)...
Skipping Git submodules setup
Executing "step_script" stage of the job script                        00:00
$ verilator --lint-only $(find . -name '*.sv')
%Error: Annex_E_Example/MEMSRAM_1024X32.sv:10:18: syntax error, unexpected ';', expecting ')'
   10 |   output [33:0] Q;
      |                 ^
%Error: UPF_Packages/UPF_package.sv:740:21: syntax error, unexpected TYPE-IDENTIFIER
  740 |     upfBooleanT     state = 0;
      |                     ^~~~~
%Error: Exiting due to 2 error(s)
Cleaning up project directory and file based variables                 00:01
ERROR: Job failed: command terminated with exit code 1
```

Duration: 7 seconds
Finished: 2 hours ago
Queued: 11 seconds
Timeout: 1h (from project)
Runner: #416 (zqgNETsBk) gitlab-runner-c66bd6b64-4zxph

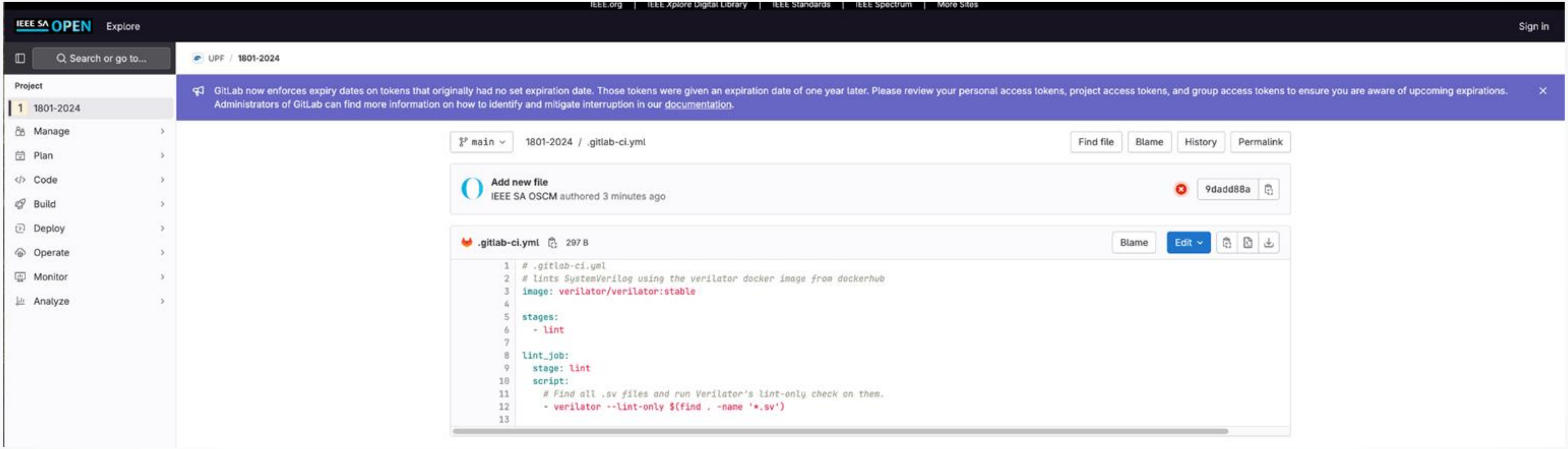
Commit 9dadd88a
Add new file

Pipeline #3873 Failed for main

lint

Related jobs

→ lint_job



GitLab now enforces expiry dates on tokens that originally had no set expiration date. Those tokens were given an expiration date of one year later. Please review your personal access tokens, project access tokens, and group access tokens to ensure you are aware of upcoming expirations. Administrators of GitLab can find more information on how to identify and mitigate interruption in our documentation.

Add new file
IEEE SA OSCM authored 3 minutes ago

```
# .gitlab-ci.yml
# lints SystemVerilog using the verilator docker image from dockerhub
image: verilator/verilator:stable

stages:
  - lint

lint_job:
  stage: lint
  script:
    # Find all .sv files and run Verilator's lint-only check on them.
    - verilator --lint-only $(find . -name '*.sv')

```

Open Source in IEEE Standards

How Open Source is Used in Standards

Case Study and Examples: IEEE 802.3 Channel Operating Margin (COM) Open Source

How Open Source is Used in Standards

Ways to create a link between a standard and open source project

1. Open source is referenced informatively
2. Open source is referenced normatively
3. Open source is not referenced in the standard but is related to it. Typical open source related to standards:
   - Test suite (software)
   - Reference Implementation (software)
   - Schema
   - Database

Note: When referencing open source in a standard, the standard can reference a *specific* version or the *latest* version.

How Open Source is Used in Standards

Examples from IEEE SA Open

- IEEE 1076-2019: VHDL Packages (https://opensource.ieee.org/vasg/Packages)
- IEEE P1159.3: PQDIF (https://opensource.ieee.org/pqdif)
- IEEE P3195.1: Ontologies Repo. (https://opensource.ieee.org/cco/CommonCoreOntologies)
- IEEE 2664-2024: Reference Implementation (https://opensource.ieee.org/sttp)
- IEEE 9274.1.1.-2023: An open source standard. https://opensource.ieee.org/xapi/about
- CIGRE Power Transformer Database – prototype & MPAI Reference Software for Neural Network Watermarking: Examples of open source related to an IEEE standard.
- IEEE 802.3 Channel Operating Margin (COM) Code (Up next)

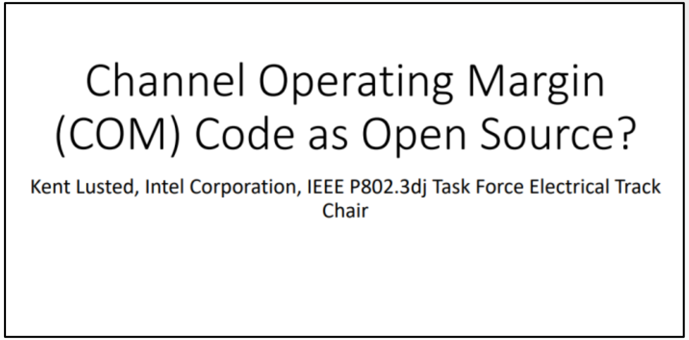
IEEE 802.3 Channel Operating Margin (COM) Open Source

Background

- IEEE Std. 802.3 and amendments normatively specify **Channel Operating Margin (COM)** via equations and methods in Annex 93A and 178A
  - Channel Operating Margin (COM) is a figure of merit for a channel derived from a measurement of its scattering parameters (S-parameters).
- There have been and continue to be contributions of software code implementations of these COM equations and methods for participants to use
  - The contributed COM software code implementation is widely used by industry participants and other standards setting organizations
- It was increasingly important to ensure that the “example” code implementation is revision controlled, peer reviewed, cross checked, and bug free and maintained over time
- The IEEE SA Open Source platform provided a viable platform for the development of the code in an open source manner that benefits broad industry

Strong Consensus Is Key

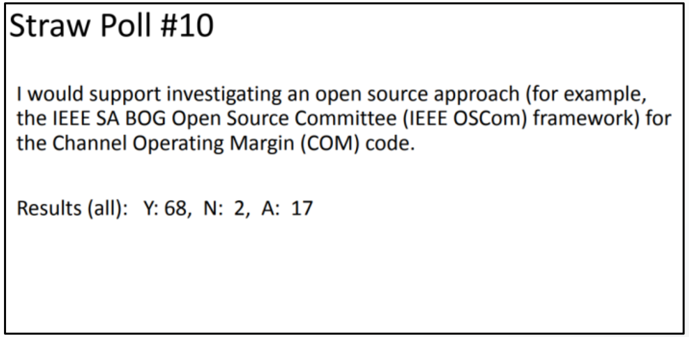
- There was strong consensus within the P802.3dj Task Force and IEEE 802.3 Working Group to investigate an open source approach for the COM code
- Much of the Q&A from the participants asked about the next level of details
  - What repo will be used?
  - How are changes processed?
  - Who will review the code?
  - Etc.



Channel Operating Margin (COM) Code as Open Source?

Kent Lusted, Intel Corporation, IEEE P802.3dj Task Force Electrical Track Chair

https://www.ieee802.org/3/dj/public/24_03/lusted_3dj_05_2403.pdf



Straw Poll #10

I would support investigating an open source approach (for example, the IEEE SA BOG Open Source Committee (IEEE OSCom) framework) for the Channel Operating Margin (COM) code.

Results (all): Y: 68, N: 2, A: 17

https://www.ieee802.org/3/dj/public/24_03/motions_3cwdj_2403.pdf

Workflow Flexibility

- There are three predominant workflows of the COM code that needed to be considered
  - Development - Fast and flexible
    - New features and new capabilities for IEEE 802.3 TF/SG use
    - Align with changes to draft specification, as the spec changes
  - Maintenance - Structured
    - Corrections to existing functions or code related to IEEE 802.3 Std.
    - Stable and “proven” releases
  - Adjacencies
    - Requests for features and capabilities beyond the IEEE 802.3 Std. (e.g. OIF, etc.)

IEEE SA Open Source supports all of these!

Key Learnings on Open Source

- Key responsibilities to assign:
  - Source Project Lead ("Chair") : Overall responsible for the OS project
  - Maintainer(s) ("Editors"): Authority to commit (save changes) to the IEEE code and document repository
- Contributors to the Open Source project will need to submit an IEEE Contributor License Agreement (CLA)
  - We selected BSD-3 clause license agreement (Requested by the contributor of the initial code commit)
- Determine the Project Tier (3/4) up front – Sets the process flow and requirements
  - Took us time to clarify that the existing COM code and configuration spreadsheets do not meet the requirement of "incorporated into IEEE standards": Moved from Tier 4 -> Tier 3
    - The COM code is not currently normatively or informatively included nor referenced as part of the text of the standard or cited in the IEEE Std. 802.3 nor amendments (e.g. 3dj).
- Strong support from IEEE SA OSCom team
- Our team is learning as we go along ☺

Summary

- IEEE SA Open Source platform is a great solution for the COM code
  - Lots of tools
  - Flexible structure
  - Great support
- COM Open Source Tier 3 project was approved by IEEE SA OSCom on 19 December 2024!
- Work currently in progress to:
  - Create the COM code repository on Gitlab
  - Perform initial COM code commit to repo
  - Establish governance related processes

SA Open

Tour of the Site

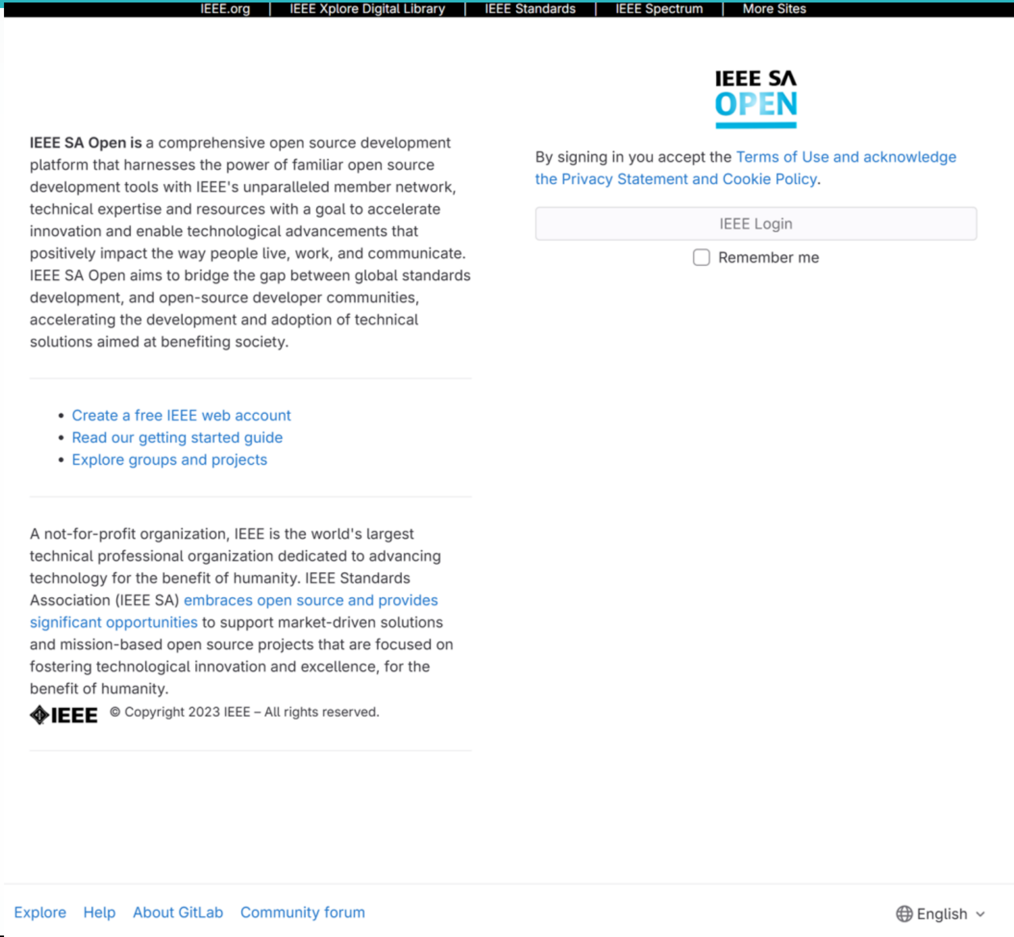
Staging vs Production

Governance and Management

SA Open

A quick tour of the site

- URL: https://opensource.ieee.org
- Must:
  - Have IEEE Account (free to anyone)
  - Agree Terms and Conditions
  - Submit CLA to make a contribution



IEEE.org | IEEE Xplore Digital Library | IEEE Standards | IEEE Spectrum | More Sites
IEEE SA OPEN
IEEE Login
Explore
Help
About GitLab
Community forum
English

IEEE SA Open is a comprehensive open source development platform that harnesses the power of familiar open source development tools with IEEE's unparalleled member network, technical expertise and resources with a goal to accelerate innovation and enable technological advancements that positively impact the way people live, work, and communicate. IEEE SA Open aims to bridge the gap between global standards development, and open-source developer communities, accelerating the development and adoption of technical solutions aimed at benefiting society.

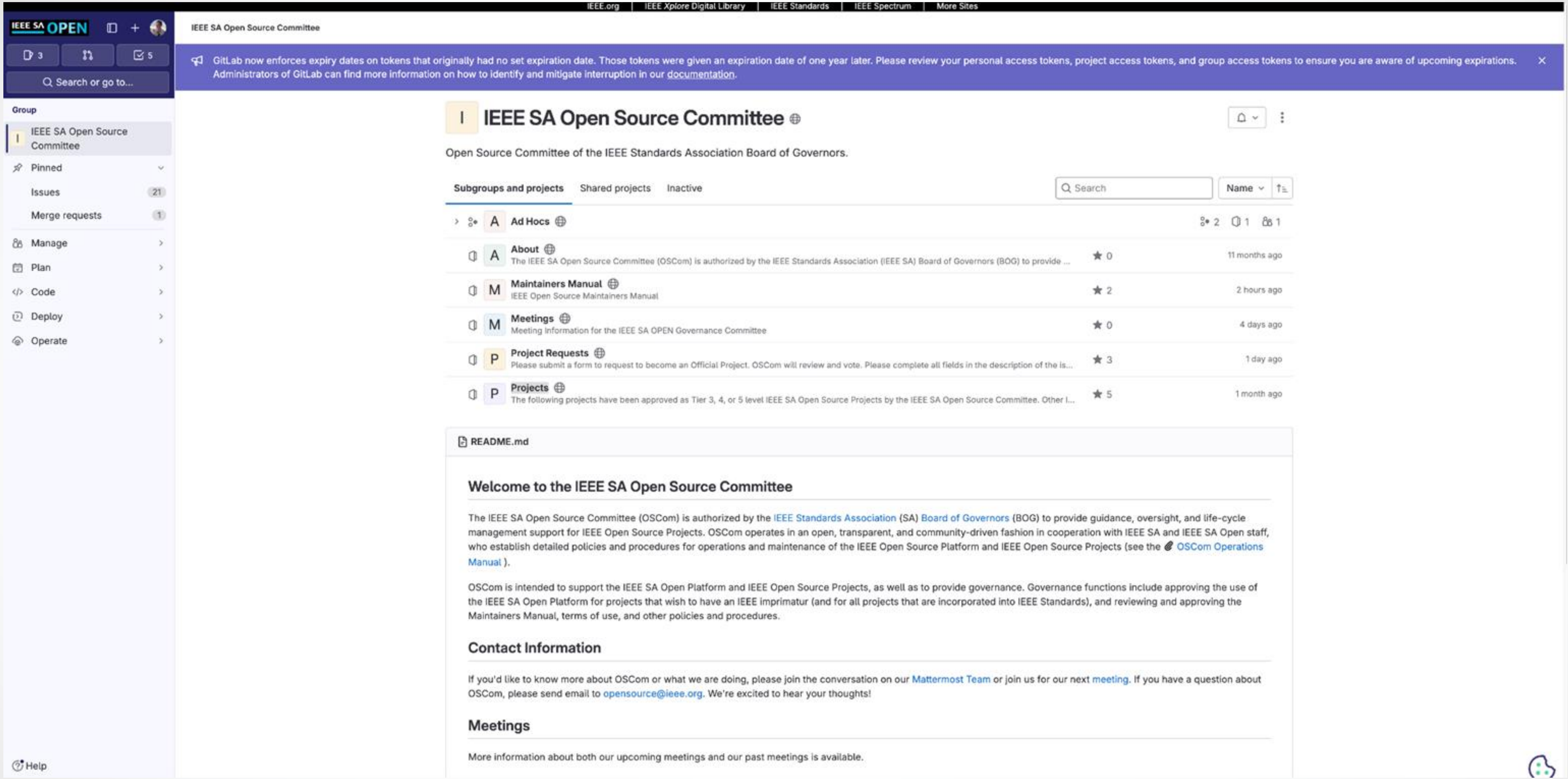
- Create a free IEEE web account
- Read our getting started guide
- Explore groups and projects

A not-for-profit organization, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity. IEEE Standards Association (IEEE SA) embraces open source and provides significant opportunities to support market-driven solutions and mission-based open source projects that are focused on fostering technological innovation and excellence, for the benefit of humanity.

IEEE © Copyright 2023 IEEE – All rights reserved.

By signing in you accept the Terms of Use and acknowledge the Privacy Statement and Cookie Policy.

☐ Remember me



GitLab now enforces expiry dates on tokens that originally had no set expiration date. Those tokens were given an expiration date of one year later. Please review your personal access tokens, project access tokens, and group access tokens to ensure you are aware of upcoming expirations. Administrators of GitLab can find more information on how to identify and mitigate interruption in our documentation.

IEEE SA Open Source Committee

Open Source Committee of the IEEE Standards Association Board of Governors.

Ad Hocs		2 · 1 · 1	
About	The IEEE SA Open Source Committee (OSCom) is authorized by the IEEE Standards Association (IEEE SA) Board of Governors (BOG) to provide ...	★ 0	11 months ago
Maintainers Manual	IEEE Open Source Maintainers Manual	★ 2	2 hours ago
Meetings	Meeting information for the IEEE SA OPEN Governance Committee	★ 0	4 days ago
Project Requests	Please submit a form to request to become an Official Project. OSCom will review and vote. Please complete all fields in the description of the is...	★ 3	1 day ago
Projects	The following projects have been approved as Tier 3, 4, or 5 level IEEE SA Open Source Projects by the IEEE SA Open Source Committee. Other I...	★ 5	1 month ago

Welcome to the IEEE SA Open Source Committee

The IEEE SA Open Source Committee (OSCom) is authorized by the IEEE Standards Association (SA) Board of Governors (BOG) to provide guidance, oversight, and life-cycle management support for IEEE Open Source Projects. OSCom operates in an open, transparent, and community-driven fashion in cooperation with IEEE SA and IEEE SA Open staff, who establish detailed policies and procedures for operations and maintenance of the IEEE Open Source Platform and IEEE Open Source Projects (see the OSCom Operations Manual).

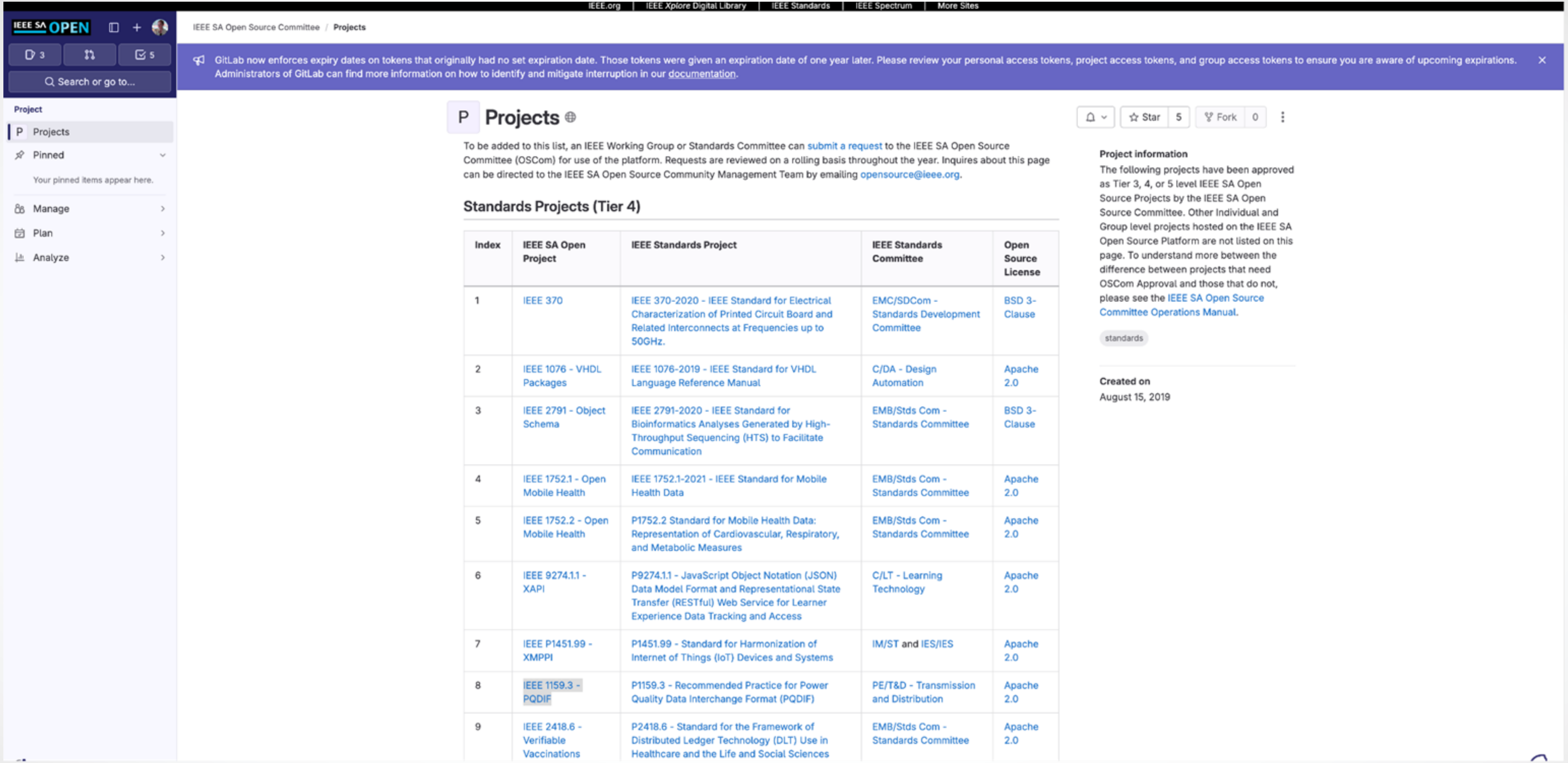
OSCom is intended to support the IEEE SA Open Platform and IEEE Open Source Projects, as well as to provide governance. Governance functions include approving the use of the IEEE SA Open Platform for projects that wish to have an IEEE imprimatur (and for all projects that are incorporated into IEEE Standards), and reviewing and approving the Maintainers Manual, terms of use, and other policies and procedures.

Contact Information

If you'd like to know more about OSCom or what we are doing, please join the conversation on our Mattermost Team or join us for our next meeting. If you have a question about OSCom, please send email to opensource@ieee.org. We're excited to hear your thoughts!

Meetings

More information about both our upcoming meetings and our past meetings is available.


IEEE.org | IEEE Xplore Digital Library | IEEE Standards | IEEE Spectrum | More Sites
IEEE SA OPEN
3
5
Search or go to...
Project
Projects
Pinned
Your pinned items appear here.
Manage
Plan
Analyze
IEEE SA Open Source Committee / Projects
Star 5
Fork 0

GitLab now enforces expiry dates on tokens that originally had no set expiration date. Those tokens were given an expiration date of one year later. Please review your personal access tokens, project access tokens, and group access tokens to ensure you are aware of upcoming expirations. Administrators of GitLab can find more information on how to identify and mitigate interruption in our documentation.

Projects

To be added to this list, an IEEE Working Group or Standards Committee can submit a request to the IEEE SA Open Source Committee (OSCom) for use of the platform. Requests are reviewed on a rolling basis throughout the year. Inquires about this page can be directed to the IEEE SA Open Source Community Management Team by emailing opensource@ieee.org.

Standards Projects (Tier 4)

Index	IEEE SA Open Project	IEEE Standards Project	IEEE Standards Committee	Open Source License
1	IEEE 370	IEEE 370-2020 - IEEE Standard for Electrical Characterization of Printed Circuit Board and Related Interconnects at Frequencies up to 50GHz.	EMC/SDCom - Standards Development Committee	BSD 3-Clause
2	IEEE 1076 - VHDL Packages	IEEE 1076-2019 - IEEE Standard for VHDL Language Reference Manual	C/DA - Design Automation	Apache 2.0
3	IEEE 2791 - Object Schema	IEEE 2791-2020 - IEEE Standard for Bioinformatics Analyses Generated by High-Throughput Sequencing (HTS) to Facilitate Communication	EMB/Stds Com - Standards Committee	BSD 3-Clause
4	IEEE 1752.1 - Open Mobile Health	IEEE 1752.1-2021 - IEEE Standard for Mobile Health Data	EMB/Stds Com - Standards Committee	Apache 2.0
5	IEEE 1752.2 - Open Mobile Health	P1752.2 Standard for Mobile Health Data: Representation of Cardiovascular, Respiratory, and Metabolic Measures	EMB/Stds Com - Standards Committee	Apache 2.0
6	IEEE 9274.1.1 - XAPI	P9274.1.1 - JavaScript Object Notation (JSON) Data Model Format and Representational State Transfer (RESTful) Web Service for Learner Experience Data Tracking and Access	C/LT - Learning Technology	Apache 2.0
7	IEEE P1451.99 - XMPPI	P1451.99 - Standard for Harmonization of Internet of Things (IoT) Devices and Systems	IM/ST and IES/IES	Apache 2.0
8	IEEE 1159.3 - PQDIF	P1159.3 - Recommended Practice for Power Quality Data Interchange Format (PQDIF)	PE/T&D - Transmission and Distribution	Apache 2.0
9	IEEE 2418.6 - Verifiable Vaccinations	P2418.6 - Standard for the Framework of Distributed Ledger Technology (DLT) Use in Healthcare and the Life and Social Sciences	EMB/Stds Com - Standards Committee	Apache 2.0

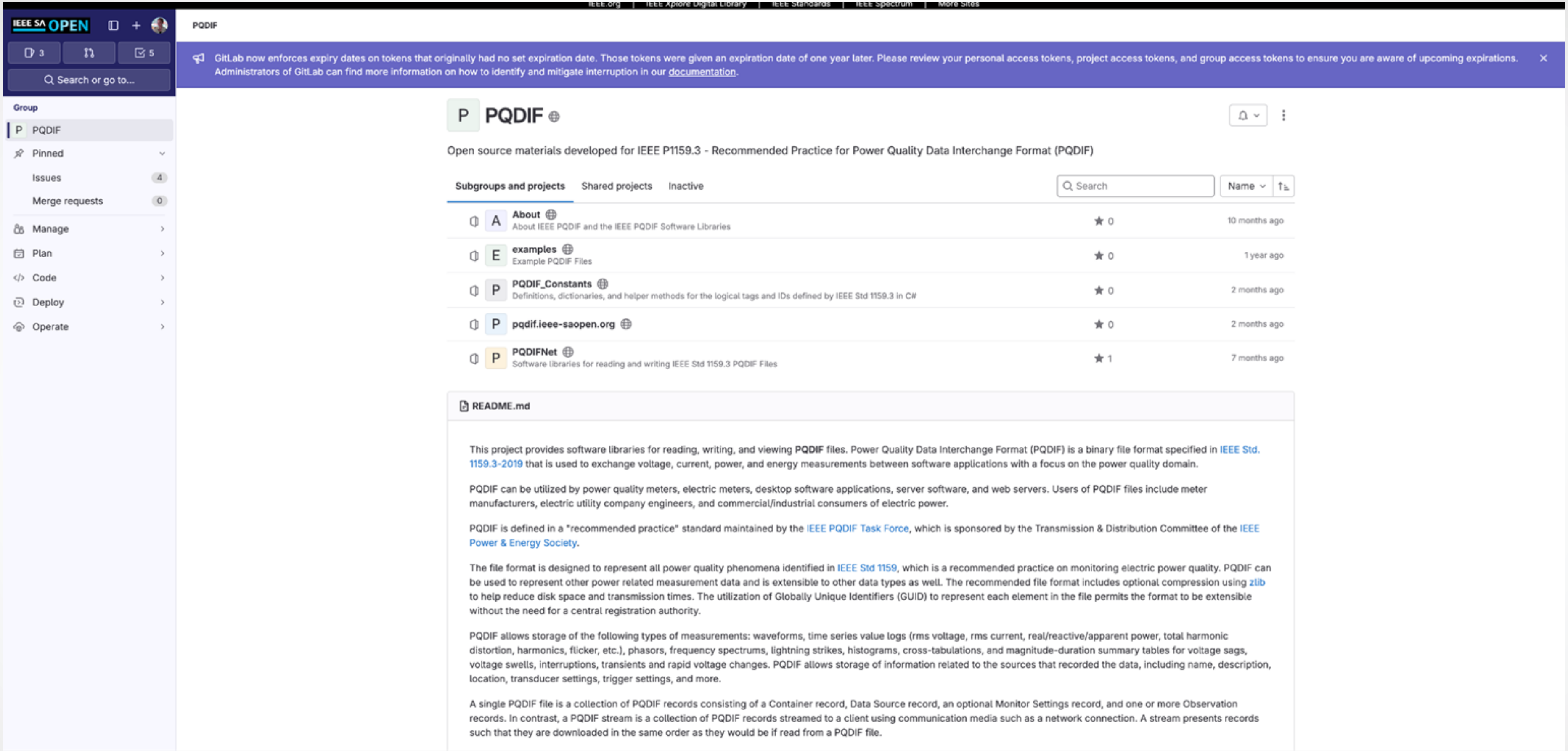
Project information

The following projects have been approved as Tier 3, 4, or 5 level IEEE SA Open Source Projects by the IEEE SA Open Source Committee. Other Individual and Group level projects hosted on the IEEE SA Open Source Platform are not listed on this page. To understand more between the difference between projects that need OSCom Approval and those that do not, please see the IEEE SA Open Source Committee Operations Manual.

standards

Created on

August 15, 2019



GitLab now enforces expiry dates on tokens that originally had no set expiration date. Those tokens were given an expiration date of one year later. Please review your personal access tokens, project access tokens, and group access tokens to ensure you are aware of upcoming expirations. Administrators of GitLab can find more information on how to identify and mitigate interruption in our documentation.

PQDIF

Open source materials developed for IEEE P1159.3 - Recommended Practice for Power Quality Data Interchange Format (PQDIF)

A	**About** About IEEE PQDIF and the IEEE PQDIF Software Libraries	★ 0	10 months ago
E	**examples** Example PQDIF Files	★ 0	1 year ago
P	**PQDIF_Constants** Definitions, dictionaries, and helper methods for the logical tags and IDs defined by IEEE Std 1159.3 in C#	★ 0	2 months ago
P	**pqdif.ieee-saopen.org**	★ 0	2 months ago
P	**PQDIFNet** Software libraries for reading and writing IEEE Std 1159.3 PQDIF Files	★ 1	7 months ago

README.md

This project provides software libraries for reading, writing, and viewing **PQDIF** files. Power Quality Data Interchange Format (PQDIF) is a binary file format specified in IEEE Std. 1159.3-2019 that is used to exchange voltage, current, power, and energy measurements between software applications with a focus on the power quality domain.

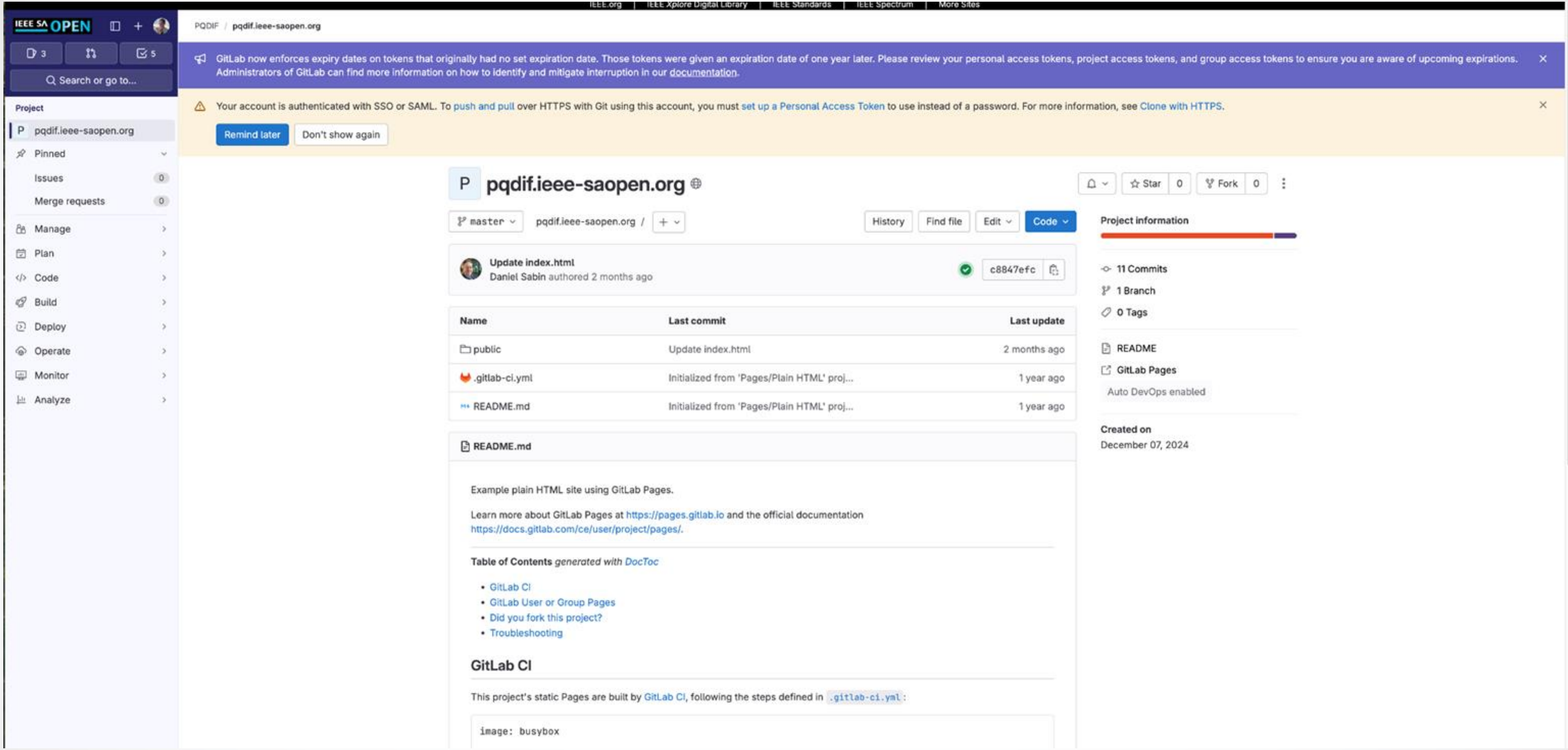
PQDIF can be utilized by power quality meters, electric meters, desktop software applications, server software, and web servers. Users of PQDIF files include meter manufacturers, electric utility company engineers, and commercial/industrial consumers of electric power.

PQDIF is defined in a "recommended practice" standard maintained by the IEEE PQDIF Task Force, which is sponsored by the Transmission & Distribution Committee of the IEEE Power & Energy Society.

The file format is designed to represent all power quality phenomena identified in IEEE Std 1159, which is a recommended practice on monitoring electric power quality. PQDIF can be used to represent other power related measurement data and is extensible to other data types as well. The recommended file format includes optional compression using zlib to help reduce disk space and transmission times. The utilization of Globally Unique Identifiers (GUID) to represent each element in the file permits the format to be extensible without the need for a central registration authority.

PQDIF allows storage of the following types of measurements: waveforms, time series value logs (rms voltage, rms current, real/reactive/apparent power, total harmonic distortion, harmonics, flicker, etc.), phasors, frequency spectrums, lightning strikes, histograms, cross-tabulations, and magnitude-duration summary tables for voltage sags, voltage swells, interruptions, transients and rapid voltage changes. PQDIF allows storage of information related to the sources that recorded the data, including name, description, location, transducer settings, trigger settings, and more.

A single PQDIF file is a collection of PQDIF records consisting of a Container record, Data Source record, an optional Monitor Settings record, and one or more Observation records. In contrast, a PQDIF stream is a collection of PQDIF records streamed to a client using communication media such as a network connection. A stream presents records such that they are downloaded in the same order as they would be if read from a PQDIF file.



GitLab now enforces expiry dates on tokens that originally had no set expiration date. Those tokens were given an expiration date of one year later. Please review your personal access tokens, project access tokens, and group access tokens to ensure you are aware of upcoming expirations. Administrators of GitLab can find more information on how to identify and mitigate interruption in our documentation.

Your account is authenticated with SSO or SAML. To push and pull over HTTPS with Git using this account, you must set up a Personal Access Token to use instead of a password. For more information, see Clone with HTTPS.

pqdif.ieee-saopen.org

Update index.html
Daniel Sabin authored 2 months ago c8847efc

Name	Last commit	Last update
public	Update index.html	2 months ago
.gitlab-ci.yml	Initialized from 'Pages/Plain HTML' proj...	1 year ago
README.md	Initialized from 'Pages/Plain HTML' proj...	1 year ago

Created on
December 07, 2024

README.md

Example plain HTML site using GitLab Pages.

Learn more about GitLab Pages at https://pages.gitlab.io and the official documentation https://docs.gitlab.com/ce/user/project/pages/.

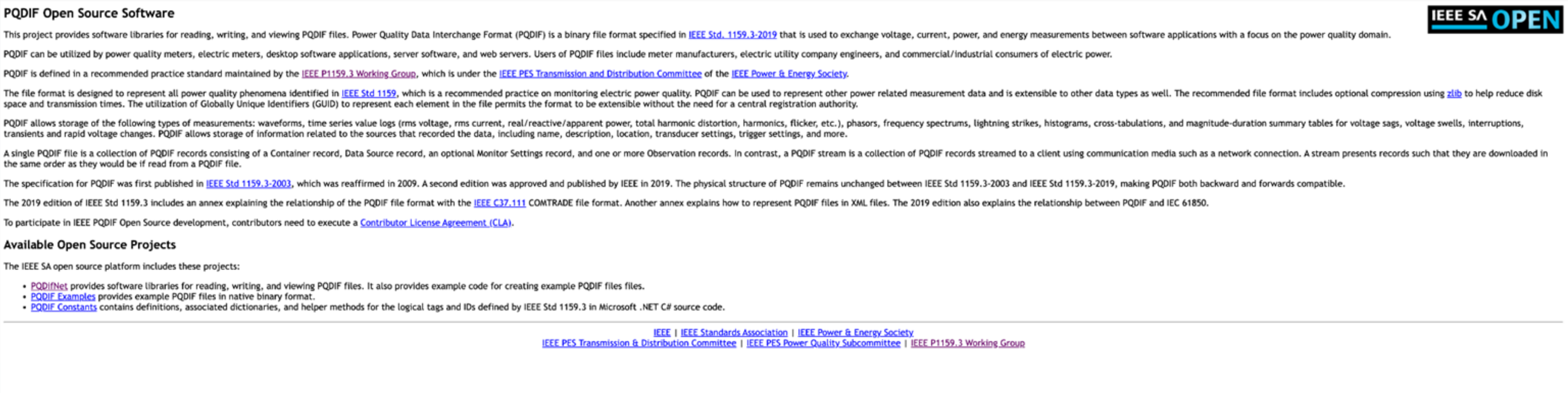
Table of Contents *generated with DocToc*

- GitLab CI
- GitLab User or Group Pages
- Did you fork this project?
- Troubleshooting

GitLab CI

This project's static Pages are built by GitLab CI, following the steps defined in `.gitlab-ci.yml`:

```
image: busybox
```



PQDIF Open Source Software

IEEE SA OPEN

This project provides software libraries for reading, writing, and viewing PQDIF files. Power Quality Data Interchange Format (PQDIF) is a binary file format specified in IEEE Std. 1159.3-2019 that is used to exchange voltage, current, power, and energy measurements between software applications with a focus on the power quality domain.

PQDIF can be utilized by power quality meters, electric meters, desktop software applications, server software, and web servers. Users of PQDIF files include meter manufacturers, electric utility company engineers, and commercial/industrial consumers of electric power.

PQDIF is defined in a recommended practice standard maintained by the IEEE P1159.3 Working Group, which is under the IEEE PES Transmission and Distribution Committee of the IEEE Power & Energy Society.

The file format is designed to represent all power quality phenomena identified in IEEE Std 1159, which is a recommended practice on monitoring electric power quality. PQDIF can be used to represent other power related measurement data and is extensible to other data types as well. The recommended file format includes optional compression using zlib to help reduce disk space and transmission times. The utilization of Globally Unique Identifiers (GUID) to represent each element in the file permits the format to be extensible without the need for a central registration authority.

PQDIF allows storage of the following types of measurements: waveforms, time series value logs (rms voltage, rms current, real/reactive/apparent power, total harmonic distortion, harmonics, flicker, etc.), phasors, frequency spectrums, lightning strikes, histograms, cross-tabulations, and magnitude-duration summary tables for voltage sags, voltage swells, interruptions, transients and rapid voltage changes. PQDIF allows storage of information related to the sources that recorded the data, including name, description, location, transducer settings, trigger settings, and more.

A single PQDIF file is a collection of PQDIF records consisting of a Container record, Data Source record, an optional Monitor Settings record, and one or more Observation records. In contrast, a PQDIF stream is a collection of PQDIF records streamed to a client using communication media such as a network connection. A stream presents records such that they are downloaded in the same order as they would be if read from a PQDIF file.

The specification for PQDIF was first published in IEEE Std 1159.3-2003, which was reaffirmed in 2009. A second edition was approved and published by IEEE in 2019. The physical structure of PQDIF remains unchanged between IEEE Std 1159.3-2003 and IEEE Std 1159.3-2019, making PQDIF both backward and forwards compatible.

The 2019 edition of IEEE Std 1159.3 includes an annex explaining the relationship of the PQDIF file format with the IEEE C37.111 COMTRADE file format. Another annex explains how to represent PQDIF files in XML files. The 2019 edition also explains the relationship between PQDIF and IEC 61850.

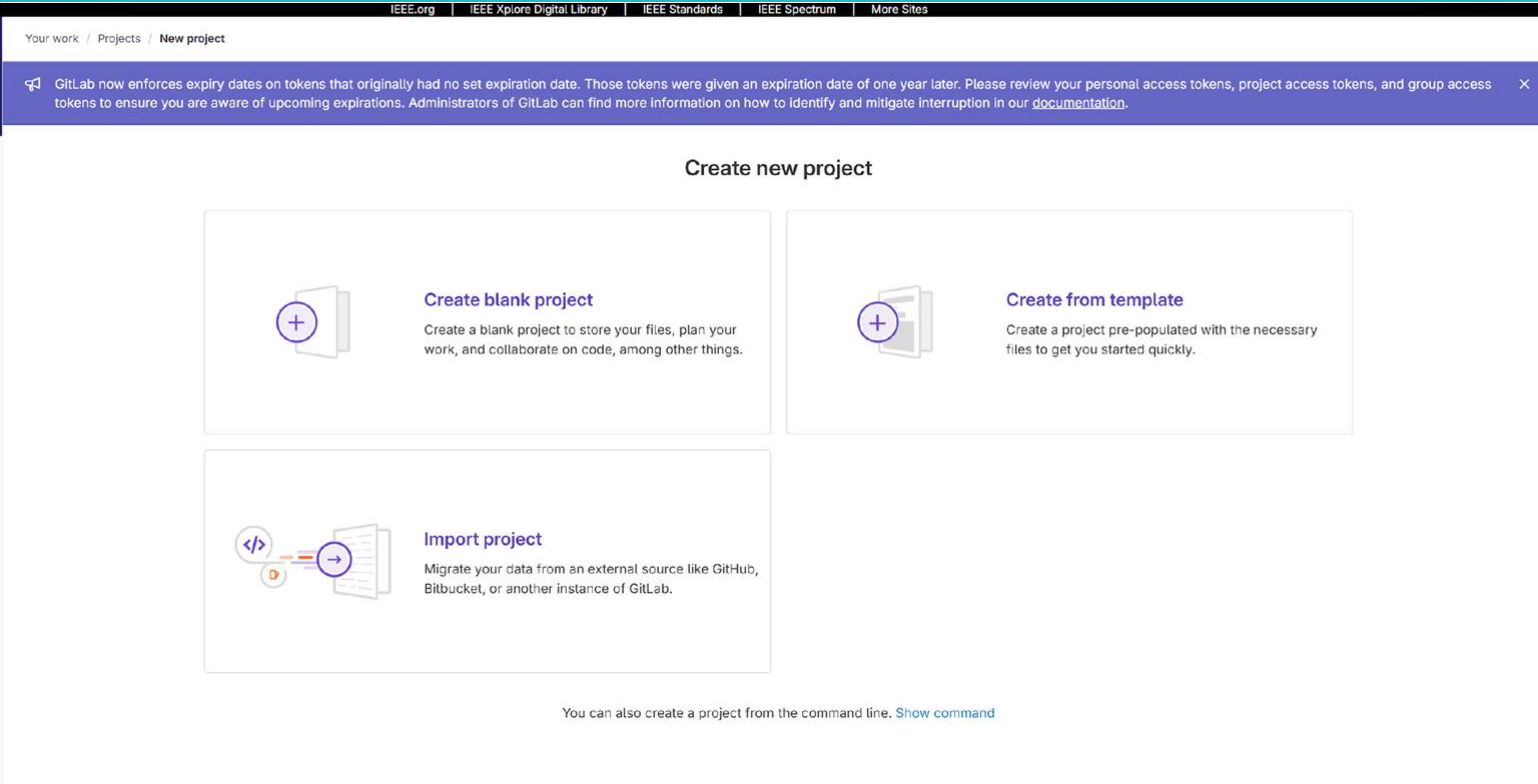
To participate in IEEE PQDIF Open Source development, contributors need to execute a Contributor License Agreement (CLA).

Available Open Source Projects

The IEEE SA open source platform includes these projects:

- PQDifNet provides software libraries for reading, writing, and viewing PQDIF files. It also provides example code for creating example PQDIF files files.
- PQDIF Examples provides example PQDIF files in native binary format.
- PQDIF Constants contains definitions, associated dictionaries, and helper methods for the logical tags and IDs defined by IEEE Std 1159.3 in Microsoft .NET C# source code.

IEEE | IEEE Standards Association | IEEE Power & Energy Society
IEEE PES Transmission & Distribution Committee | IEEE PES Power Quality Subcommittee | IEEE P1159.3 Working Group


IEEE.org | IEEE Xplore Digital Library | IEEE Standards | IEEE Spectrum | More Sites
Your work / Projects / New project

GitLab now enforces expiry dates on tokens that originally had no set expiration date. Those tokens were given an expiration date of one year later. Please review your personal access tokens, project access tokens, and group access tokens to ensure you are aware of upcoming expirations. Administrators of GitLab can find more information on how to identify and mitigate interruption in our documentation.

Create new project

Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Create from template

Create a project pre-populated with the necessary files to get you started quickly.

Import project

Migrate your data from an external source like GitHub, Bitbucket, or another instance of GitLab.

You can also create a project from the command line. Show command

Staging vs Production

Providing Standards Committees or Working Groups with a Review Mechanism

Problem: Standards Committees or Working Groups may want to review code or other open source objects prior to pushing them to the area intended to be used for download to check for possible external licensing, commercial terms, quality, and other issues.

Solution: Two projects, one under the Staging Namespace, one under the Production Namespace.

Note: Both are open source, so cannot prevent Staging instances from being forked, but disclaimers indicate that they should not be.

Governance and Management

SA Open Tiers for standards and standards related projects

Governance of Tier 3 projects

Governance of Tier 4 projects

SA Open Project Request Form

Key Links:

https://standards.ieee.org/wp-content/uploads/import/documents/other/OSCOM_Operations_Manual.pdf

https://standards.ieee.org/wp-content/uploads/import/documents/other/sb_om.pdf

https://opensource.ieee.org/oscom/maintain

SA Open Tiers for standards and standards related projects

From the OSCom Operations Manual

- **Tier 3** — Open Source Projects reviewed and approved for use of the IEEE Open Source Platform by OSCom to create IEEE Open Source Releases or products.
- **Tier 4** — IEEE Open Source Projects incorporated into IEEE standards—IEEE Open Source Projects operating in conjunction with an SASB authorized standards Project.

Governance of Tier 3 Projects

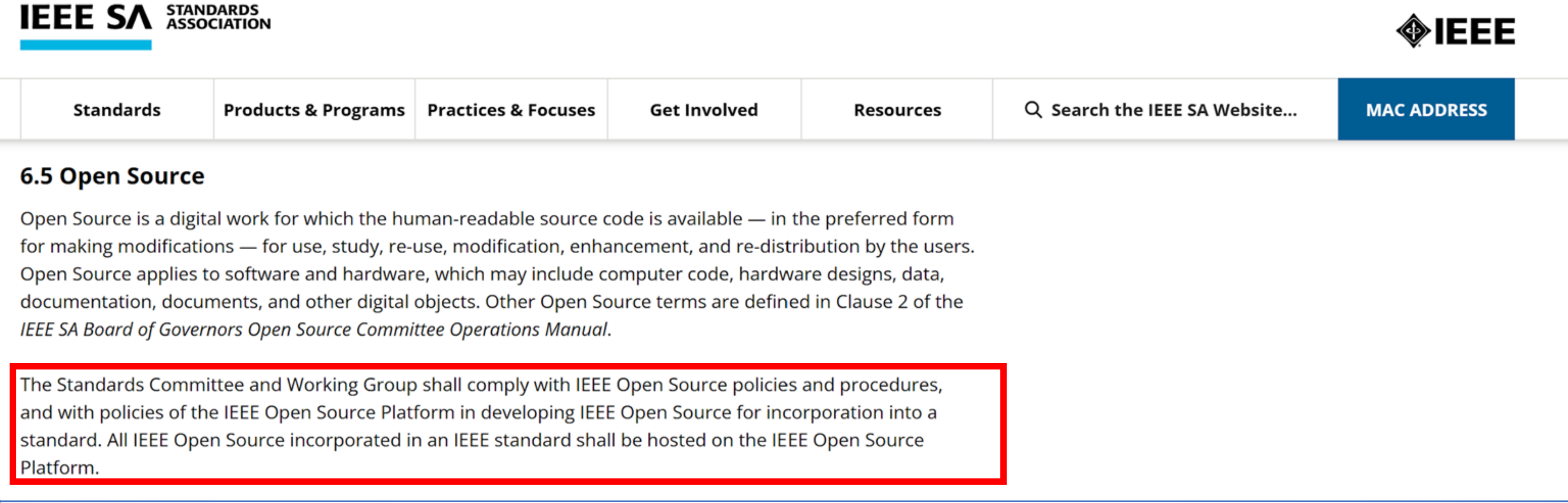
SA Open Governance Templates are Available

- Within the framework of IEEE policy, Tier 3 groups decide on the governance of their projects
- IEEE SA Open has GOVERNANCE.md templates that can be used (templates are being updated)
- The Community Manager can assist the group as they determine an appropriate governance structure
- Tier 3 projects that are intended to be incorporated into an IEEE Standards Project in the future should consider using governance similar to that for Tier 4 projects

https://opensource.ieee.org/community/resources/-/blob/main/File%20Templates/GOVERNANCE.md

Governance of Tier 4 Projects

Set out in Section 6.5 of the IEEE SA Standards Board Operations Manual



6.5 Open Source

Open Source is a digital work for which the human-readable source code is available — in the preferred form for making modifications — for use, study, re-use, modification, enhancement, and re-distribution by the users. Open Source applies to software and hardware, which may include computer code, hardware designs, data, documentation, documents, and other digital objects. Other Open Source terms are defined in Clause 2 of the *IEEE SA Board of Governors Open Source Committee Operations Manual*.

The Standards Committee and Working Group shall comply with IEEE Open Source policies and procedures, and with policies of the IEEE Open Source Platform in developing IEEE Open Source for incorporation into a standard. All IEEE Open Source incorporated in an IEEE standard shall be hosted on the IEEE Open Source Platform.

Highlights of Section 6.5 (Applies to Tier 4)

- Project requires approval of OSCom. Discuss with community manager first.
- Shall use single approved license (BSD 3-clause or Apache 2.0)
- Can use generic CLA that points to license. CLAs are mandatory for contributors.
- Responsibility for development falls to the Standards Committee and its WGs
- SC or WG shall form one or more Open Source Subgroups. Each shall have an OS Project Lead who shall be an officer of the responsible SC, or if only one WG under the SC develops Open Source, can be an officer of the WG.
- Each project shall have at least one Maintainer, who shall be a member of the responsible group developing the Open source.
- The IEEE SA patent policy and IEEE SA copyright policy apply regardless of whether a CLA has been submitted.
- During SA ballot, balloters shall have access to incorporated Open Source. The Project Lead and at least one maintainer shall be members of the Comment Resolution Group.
- Normatively referenced Open Source that is undated may be updated only with approval from the Standards Committee, Working Group, or a subgroup delegated with that responsibility.
- Final release shall occur only after approval of the standard by the SASB.

Template for Applying for a Tier 3 or Tier 4 Project

IEEE SA OPEN

IEEE SA Open Source Committee (OSCom)

Open Source Project Request

Title:

OS Project Lead/POC:

Last updated January 2023

IEEE

PROJECT TITLE AND RELATED STANDARDS

Open source project title:

Related standards project (if applicable):

- **PAR number or standard number:**
- **Scope statement:**

- **Explanation of what the standard does:**

PROJECT DESCRIPTION

What open source will be developed:

Why this is valuable:

RELATION TO EXISTING OPEN SOURCE

Relation to known open source:

Description of pre-existing open source that will be used (if any):

GOVERNANCE

Requested license (and reason):

How will the project be governed?

SUMMARY

Summary of the project and your questions for OSCom:

Review of the formal project request: (To be completed by the OSCom Administrator.)

THANK YOU

https://saopen.ieee.org

Discussion

Q&A

THANK YOU (References Follow)

Contacts:

Robby Robson: robby@ieee.org

Sarah Womer: osintunleashed@gmail.com

Kent Lusted: kentlusted@ieee.org

References

IEEE SA Open

- IEEE Open Source Platform: https://opensource.ieee.org/
- List of Projects: https://opensource.ieee.org/oscom/projects
- Maintainers Manual: https://opensource.ieee.org/community/manual
- OSCom Meeting Information: https://opensource.ieee.org/oscom/meetings
- Project Request Page: https://opensource.ieee.org/oscom/official-project-requests
- Project Request Template: https://opensource.ieee.org/oscom/official-project-requests/-/raw/master/OSCom_Project_Proposal_Template.pptx?ref_type=heads

References

IEEE SA Open Approved Licenses and CLAs

- Apache 2.0 License: https://www.apache.org/licenses/LICENSE-2.0
- BSD 3-Clause License: https://opensource.org/license/bsd-3-clause
- CERN Open Hardware Licenses (not to be used for standards development): https://opensource.ieee.org/community/cla/cern-ohl
- IEEE SA Open CLAs: https://opensource.ieee.org/community/cla

Governance Documents

- OSCom Operations Manual: https://standards.ieee.org/wp-content/uploads/import/documents/other/OSCOM_Operations_Manual.pdf
- Standards Board Operations Manual Section on Open Source https://standards.ieee.org/about/policies/opman/sect6/#6.5
- Standards Association ByLaws Section on Open Source Development: https://standards.ieee.org/about/policies/bylaws/sect5/#5.6

References

General information on Open Source

- Open Source Initiative Definition of Open Source: https://opensource.org/osd
- The Open Source Way: https://www.theopensourceway.org/

Articles on Open Source and Standards

- *Standard setting organizations and open source communities: Partners or competitors?* (Article)
- Divide Narrows Between Open Source Community, Standards Bodies (Devops.com)
- *Introduction to Open Source, Open Standards and Self Describing Data* (Blog)
- *Managing API sprawl: the case for standards and consistency in API operations* (Blog from the Cloud Native Computing Foundation)