Lawrence Livermore National Laboratory

WEAVE

Newsletter #2

March 2025

Contents

Essentials	1
Around WEAVE	2
Featured: WEAVE Badging	3
Projects Updates	5
Resources	7
Pro Tips	7
Meet the Team	8
Auspices	9

WEAVE Badging Program Kicks In

WEAVE Essentials

IN THIS ISSUE

- **WEAVE Introduces its Badging Program**
- **Dan Laney to leave the Workflow Project**
- **Sina to add hdf5 output format**
- **More tutorials are being added**

WHAT YOU NEED TO KNOW

Building on the WEAVE environment, the WEAVE team officially kicked off its badging program (see featured section) which should help tool sharing and reuse accross LC.

Our first badged tool is on the RZ.

Around WEAVE

WEAVE Staff turn around

Unfortunately Dan Laney is leaving WSC/CP and the Workflow project. Dan will still be kicking around SD, helping with the Data Governance Council and working with CED on digital engineering/thread. He will have a new 50% role leading the Architecture and Infrastructure team for NAERM `https://www.energy.gov/oe/north-american-energy-resilience-model`, a cloud-based ModSim capability (managed out of GS E program). Dan's leadership and experience will be dearly missed! While the search for Dan's replacement goes on Matt Nelms will be the acting project lead.

Additionally Xiao Chen also left the WEAVE team, he will be lending his UQ expertise to the ML4C team (where we will still be able to interact).

On happier news, Barry Sly-Delgado is slated to come back this summer for another DSTI internship on Merlin and Taskvine, with a focus on executing subgraphs of workflows. He will be joined by Marcus Hsieh, another student, as part of the Computing Internship program that will help to develop a task server interface for Merlin; allowing task server plugins to be created.

We are looking forward to hosting them this summer.

WEAVE Collaborations

Programmatically WEAVE tightened its collaboration with

- WPD modeling suites, where the WEAVE tool integration will be tightened and expanded.
- Started to provide support to the ML4C team with the goal to productize some of their tools via the Badging Program
- Engaged with the FPI and W87-1 teams in regards to their UQ tools.
- Joined the MADA effort which among other things is training an LLM to autogenerate Merlin yaml files.

FEATURED ITEM: WEAVE Badging Program

The WEAVE environment (see Newsletter #1) provides a robust solution for users to focus on their work rather than the complexities of computer science infrastructure.

While the WEAVE environment comes preloaded with many frequently used Python tools as well as all of the WEAVE tools, there has historically been no streamlined way for teams to share tools developed within a single ecosystem.

Recognizing the need and benefits for teams to share their tools and discover others' tools, WEAVE introduced the WEAVE Badging Program.

What is WEAVE Badging?

At its core, the WEAVE Badging Program aims to allow teams to share and promote their tools with others via the WEAVE environment without disruption of their current processes. The Badging program is designed to disseminate reliable, well-tested tools that can benefit teams beyond the original developers.

The program ensures that badged tools are integrated into the WEAVE environment and tested nightly to guarantee reliability. This rigorous testing process ensures that end users always have access to a functional and up-to-date version of the tool.

Additionally, badged tools can be restricted to specific subsets of users via UNIX groups, providing flexibility for developers who may wish to limit access to certain audiences.

What are the benefits of badging a tool?

Badging a tool provides numerous advantages for developers and the broader WEAVE community:

- **Increased Exposure:** Your tool gains visibility within the WEAVE ecosystem, making it easier for others to discover and adopt.
- **Reduced Effort Duplication:** By sharing your tool, others can build on your work rather than reinventing the wheel, saving time and resources.
- **Community Contributions:** With the increased visibility of your tool, it may promote community to contribute improvements, bug fixes and new features.
- **Justification for Development Time:** A popular tool can make it easier to justify spending time on further development and polishing, potentially securing dedicated resources or funding.
- **Simplified Portability:** Since badged tools are integrated into the WEAVE environment, dependencies are managed for you, making it easier to port the tool to new systems.
- **Integration Support:** The WEAVE team can provide limited support for software engineering and integration within the WEAVE framework (e.g., adding Kosh operators or orchestration features).

Are there security concerns?

Developers retain full control over the lifecycle of their tools, including access restrictions. By default, badged tools are available to all users on the relevant zone (CZ/RZ/SCF). However, developers can restrict access to specific UNIX groups (e.g., `us_cit`), ensuring that deployment is limited to authorized users only.

This flexibility allows developers to balance the benefits of sharing their tools with the need for security and control.

What are the requirements for badging a tool?

To qualify for the WEAVE Badging Program, a tool must meet the following criteria:

- **Broad Interest:** The tool should be generic enough to be of interest to a larger group of users beyond the original developers.
- **Test Suite:** The tool must include a test suite to ensure reliability and facilitate nightly testing within the WEAVE environment.

- **Documentation:** The tool must have sufficient documentation to enable others to understand and use it effectively.
- **Pip Installable:** The latest version of the tool must be discoverable and installable via `pip` from a repository.

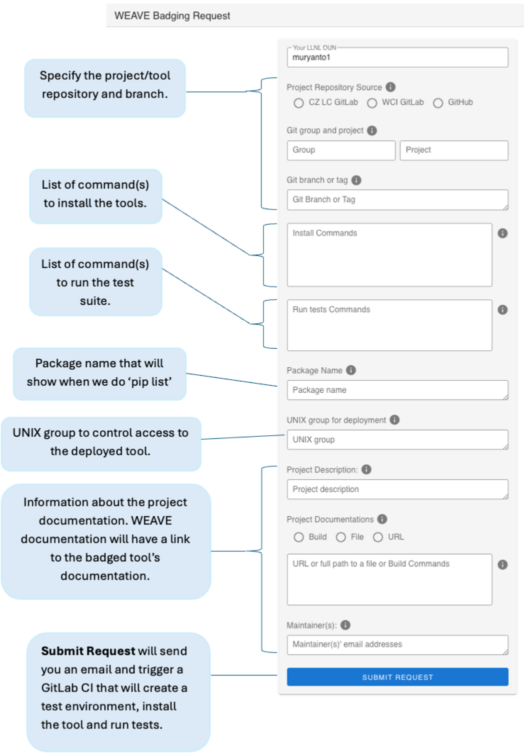
How to get started?

If you are interested in badging your tool, the WEAVE team is here to help. In addition to providing guidance on meeting the requirements, the team can offer limited support for software engineering and integration tasks, such as adding Kosh operators or other framework-specific features.

To begin the process, reach out to the WEAVE team with the following information:

- A brief description of your tool and its intended use case.
- Confirmation that the tool meets the badging requirements (or a plan to address any gaps).
- Any specific integration or support needs you may have.

When you are ready to get your tool badged, please visit the zone-specific (depending on which zone you want your tool to be deployed) WEAVE Badging Request form at CZ `https://lc.llnl.gov/weave_badging`, RZ `https://rzlc.llnl.gov/weave_badging` and SCF `https://lc.llnl.gov/weave_badging`.



WEAVE Badging Request

Your LLNL OUN: muryanto1

Project Repository Source ○ CZ LC GitLab ○ WCI GitLab ○ GitHub

Git group and project: Group | Project

Git branch or tag: Git Branch or Tag

Install Commands

Run tests Commands

Package Name: Package name

UNIX group for deployment: UNIX group

Project Description: Project description

Project Documentations ○ Build ○ File ○ URL

URL or full path to a file or Build Commands

Maintainer(s): Maintainer(s)' email addresses

Specify the project/tool repository and branch.

List of command(s) to install the tools.

List of command(s) to run the test suite.

Package name that will show when we do 'pip list'

UNIX group to control access to the deployed tool.

Information about the project documentation. WEAVE documentation will have a link to the badged tool's documentation.

Submit Request will send you an email and trigger a GitLab CI that will create a test environment, install the tool and run tests.

Figure. 1: Registering your tool for badging

Conclusion

The WEAVE Badging Program is an excellent opportunity to share your tools with the broader community, gain valuable contributions from other users, and streamline your development process. By participating, you not only enhance the visibility and impact of your work but also contribute to the growth and success of the WEAVE ecosystem.

Please contact the WEAVE team to get your tool badged

WEAVE Project Updates

Maestro (white242@llnl.gov)

⚠ **Maestro dropped support for Python 3.7**

You can change sleep, restart limit, and throttle settings in the middle of an executing study.

Merlin (gunnarson1@llnl.gov)

Merlin release version 1.12.2, which includes:

- ‘merlin info‘ is cleaner and gives python package info
- bug fixes for the ‘merlin status‘ command

Merlin is developing several new ways to help keep your study alive through errors related to server connections:

- updates to the ‘merlin monitor‘ command so that it will automatically restart a workflow when hanging
- a new ‘merlin manager‘ command that will watch over your Celery workers, automatically picking them back up when problems arise

Keep an eye out for API documentation being released soon!

Sina (haluska2@llnl.gov)

- In addtiion to the traditional JSON format, the Sina library can now output in HDF5 format. This opens the door to more efficient handling of appending to the files as the codes are running.
- More work was done on the Ares integration, which can now generate intermediary dumps of the full record or a subset of it. Other codes are following.
- Sina 1.15 was released.
- Work toward the Sina 2.0 Python module has started. Python 2.0 support will (**finally**) be dropped.

Kosh (moreno45@llnl.gov)

Kosh implemented many new features. These are already in develop and will be in the upcoming 3.2 release.

- Kosh now has L1, L2 and L* operators.
- Kosh ensembles can now have ensemble-level tags.
- Ensembles can be exported to a Pandas dataframe.
- Kosh now offers a lock strategy which helps when scaling Kosh store to many parallel operations.

PyDV (moreno45@llnl.gov)

PyDV fixed a bug where creating a plot log scale wasn’t working.

Many features were added as well:

- ‘disp‘ now has format options.
- ‘image‘ now has width and height options.
- Cleaned up docs for consistency.
- Consolidated ‘pydvpy.makecurve()‘ and ‘curve.Curve()‘. Now uses ‘pydvpy.makecurve()‘ throughout PyDV.
- Changed ‘from pydv import pydvpy as pydvif‘ to ‘from pydv import pydvpy‘. Now uses ‘pydvpy‘ throughout PyDV.
- ‘mathinterpparams‘: Users can now set ‘numpy.interp()‘ ‘left‘, ‘right‘, and ‘period‘ parameters for internal curve math methods for Curve such as ‘+ a b c‘, ‘- a b c‘, etc....
- Changed the default python from weave develop to prod.

WEAVE Resources

WEAVE Community

You can reach out to the WEAVE team or the WEAVE community on our MS Team or via email at: `weave-support@llnl.gov`

WEAVE Documentation

The WEAVE documentation can be found on readthedocs at: `https://llnl-weave.readthedocs.io/en/latest/index.html`

On LC you can access our development documentation on CZ, RZ, SCF at:

`https://lc.llnl.gov/weave` and `https://rzlc.llnl.gov/weave`

A set of tutorials (some accessible only from their allowed zone) can be found under the tutorials section of the above site.

WEAVE Environment

All WEAVE packages are available via pip. Most of them are also available with Spack or Conda.

On LC you can access WEAVE on CZ, RZ, SCF at:

/usr/apps/weave

e.g: **source /usr/apps/weave/weave-prod-cpu/bin/activate**

See instructions at: `https://lc.llnl.gov/weave/llnl/environment.html`

Note: If you prefer or need to create your own virtual environment you can do so from our Spack-based distribution:

/usr/apps/weave/tools/create_venv.sh -p cpu -e venv -v latest-stable

or to create from our development version:

/usr/apps/weave/tools/create_venv.sh -p cpu -e venv -v latest-develop

Pro Tips

Maestro Tip

Annoyed by the pesky timestamp on your Maestro output directory? Use the -o my_dir option to force the directory name.

WARNING: If the directory already exists it will be overwritten.

The WEAVE Team

Charles Doutriaux (Team Lead)

Charles has nearly 30 years of experience at the lab. He obtained his post-graduate degree in Climate and Physico-Chemistry of the Atmosphere at the University Joseph Fourier in Grenoble, France, in 1996. He subsequently joined LLNL's Program for Climate and Model Diagnostics where he became the technical lead for the Community Data Analysis Tools (CDAT) and was a contributing author to the 2007 Nobel Peace Prize Winning Intergovernmental Panel on Climate Change Fourth Annual Report (IPCC AR4). He later led the Earth System Grid's Compute Working Team international group. ESGF. ESGF and CDAT received multiple Federal Laboratory Consortium for Technology Transfer awards and ESGF received the 2017 R&D 100 award. In 2019, Charles joined WSC's to become the Advanced Machine Learning group (aka Vidya) deputy lead. As part of Vidya Charles created Kosh an open-source tools for data management. In 2022, Charles assumed the role of team leader for the Workflow Enablement and Advanced Environments (WEAVE), with a focus on developing tools that empower users' workflows and enhance data management solutions.

Charles Doutriaux

Brian Gunnarson (Merlin Lead, WF Orchestration)

Brian Gunnarson graduated Magna Cum Laude from the University of Oregon (UO) in 2022 with a bachelor's degree in Computer Science and minors in Mathematics and Computer Information Technology. While at UO, he co-founded Inquire, a website designed to improve communication between professors and students during the COVID-19 pandemic, which was implemented in multiple classes. Shortly after earning his degree, Brian joined the Workflow Enablement and AdVanced Environment (WEAVE) team as the software lead for Merlin, a workflow orchestration tool. During his two years at the lab, he received a WSC Silver Award for his contributions to Merlin and co-authored a paper with the Inertial Confinement on El Capitan (ICECap) team that was published in Physics of Plasmas.

Rebecca Haluska (Sina Lead, Data Management POC)

Rebecca 'Becky' Haluska has been at the lab for 7 years, working on projects from CSR and performance monitoring to ML-driven antibody design, plus outreach through GWC and the EIP. Under the auspices of WEAVE, she works with users and code teams to increase the availability of critical simulation data, as well as designing query and visualization tooling modules to help users test, build, and automate workflow data management. Most of this finds a home in Sina, an open-source tool hosted and supported by WEAVE that's seen integration with many major codes around the lab.

Jorge Moreno (Kosh and PyDV Lead, Data Management)

Jorge Moreno is the lead developer for Kosh and PyDV. He attended the University of California, Berkeley where he obtained an M.S. in Mechanical Engineering in 2017 while researching within the Combustion Modeling Lab. He joined Sandia National Laboratories as part of the Systems team in 2018 where he was the lead for multiple tests and the Modeling and Simulation liaison lead. Later on, he joined the Thermal Simulation team where he was the PI for a Verification, Validation and Uncertainty Quantification (VV&UQ) project and the lead for system level thermal analyses. He joined LLNL in August 2022. Jorge split his time between WEAVE and is embedded with other projects to help improve and develop WEAVE tools. He originally worked with WPD Suites and moved to the Vidya Advanced Machine Learning group in April 2024.

Jorge Moreno

Lina Muryanto (CI/CD Lead)

Lina Muryanto

Lina received a B.S in Mathematics with Computer Science from Massachussetts Institute of Technology, Cambridge and a M.S in Computer Science from Stanford University. She was a senior software engineer on the Solaris Cluster team at Oracle from 2000 until 2012 and the principal software engineer for the Distributed Resource Allocation Manager (DReAM) team at oracle from 2012 until 2017. In 2018 she joined LLNL as the lead computer scientist for continuous integration and development (CI/CD) on the Earth System Grid Federation Project. In 2021 Lina joined the strategic deterrent program. She joined WEAVE in 2022 and has built their CI/CD from scratch. Lina is passionate about achieving high software quality and reliability through software test development, automation of software integration and release process. Lina Muryanto is the lead for WEAVE's CI/CD and Documentation

Renee Olson (IBIS and Trata Lead, UQ/ML Integration Lead)

Renee has been at the lab for nearly three years working on various projects. She is currently the lead developer for WEAVE's UQ tools, Trata and IBIS, and is working to make more lab-created UQ and ML tools available in the WEAVE virtual environment. Her research interests include clustering, data reduction, Bayesian inference, and computer vision. During graduate school, Renee completed two summer internships at LLNL working with the DTED Flight Sciences team on a multi-fidelity Gaussian process model and a hierarchical Bayesian Inference with mentors including Ana Kupresanin, Kevin Quinlan, and Jeremy Thornock. Renee graduated with a M.S. degree in Statistics/Data Science in 2021 from Cal State East Bay, and a B.S. from Brigham Young University.

Gabriel Waegner (Graduate Program, Sina)

Gabriel Waegner is a recent Computer Science graduate from Arizona State University, working as an Academic Graduate Appointee at LLNL formerly with the ACES team as a front end developer and API tester, and is now working with the WEAVE team to optimize file management.

Jeremy White (Maestro Lead, WF Orchestration POC)

Jeremy White is a Physicist and Software Developer at Lawrence Livermore National Laboratory. His research interests include algorithm development for multi-physics simulations and modeling, computational workflow automation and infrastructure, and software engineering. He holds a Ph.D. in Engineering Mechanics from the University of Wisconsin-Madison.

Auspices

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

LLNL-BR-872736