Earth Prediction Innovation Center - Community Infrastructure

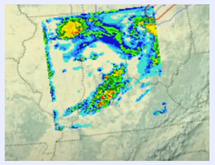
KEY PARTNERS

ACKNOWLEDGEMENT

- NOAA OAR: WPO, GSL, PSL, NSSL, CSL, AOML, GFDL
- NOAA OPEN DATA DISSEMINATION (NODD) PROGRAM
- NWS: EMC, OSTI
- DTC
- UCAR: CGD, JCSDA
- ACADEMIA: GEORGE MASON UNIVERSITY, OKLAHOMA UNIVERSITY, UNIVERSITY OF MICHIGAN
- COOPERATIVE INSTITUTES: CIRES, CIMSS
- CSPS: AWS, AZURE, AND GOOGLE CLOUD
- OCIO/GDIT/PWS

SHORT RANGE WEATHER

Current version: 2.2.0
Release Date: Oct 16, 2023



Release Updates:

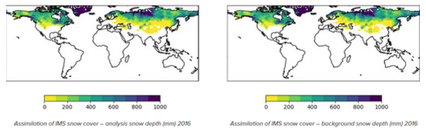
- Bug fixes
- Conversion to Python workflow
- Improved container support
- Updates to CCPP that target the top of the main branch
- Support for the UPP inline post option
- Addition of a verification package (METplus) for both deterministic and ensemble simulations
- Support for four stochastically perturbed physics schemes

LAND DATA ASSIMILATION

Current version: 1.1.0
Release Date: May 8, 2023
Release Updates:

- Land DA is an offline version of the Noah Multi-Physics land surface model used in the UFS Weather Model.
- Its data assimilation framework uses the Joint Effort for Data assimilation Integration software.

Next release date: Dec 8, 2023
Next release version: 1.2.0



Assimilation of IMS snow cover – analysis snow depth (mm) 2016 *Assimilation of IMS snow cover – background snow depth (mm) 2016*

CODE STANDARDS

EPIC is working with the community to impliment code standards and a “Clean as you code” development mentality.

	CCPP Physics	HAFS	LandDA	LandDA Workflow
Reliability	E	E	A	E
Security	A	A	A	A
Security Review	E	E	A	E
Maintainability	D	A	D	D
Bugs	141	1,300	0	132
Vulnerabilities	0	0	0	0
HotSpots	9	25	0	9
Code Smells	94,000	773,000	296	91,000
Technical Debt	5,858 Days	48,180 Days	18 Days	5,708 Days
Lines of Code	321K	2.6M	1.4K	303K

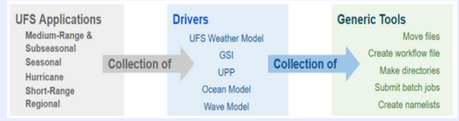
	MRW	SRW	UFS Utils	UFS WM	Workflow Tools
Reliability	A	E	E	E	C
Security	A	A	A	A	A
Security Review	E	E	E	E	E
Maintainability	A	A	D	D	A
Bugs	0	160	658	209	1
Vulnerabilities	0	0	0	0	0
HotSpots	2	10	9	20	3
Code Smells	55	3,700	126,000	443,000	55
Technical Debt	6 Hours	19 Days	7,880 Days	27,612 Days	1 Day
Lines of Code	2.7K	30K	398K	1.5M	3.8K

Quality gate “Clean as You Code” methodology:

- No new bugs are introduced.
- No new vulnerabilities are introduced.
- All new security hotspots are reviewed.
- New code has limited technical debt.
- New code has limited duplication.
- New code is properly covered by tests.

WORFLOW TOOLS

The idea of a **unified workflow** is to develop a wide array of tools in object-oriented Python that can be used in all UFS applications. These tools can be used to perform minuscule tasks such as movement of files within or across systems, manipulating datetime objects, parsing and populating configuration files, and executing and processing a series of functions in a job or a task within an application suite.



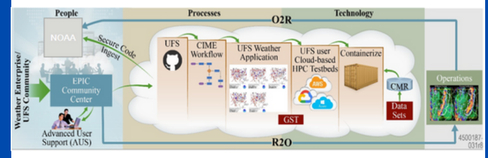
COMMUNITY INFRASTRUCTURE

EPIC provides repositories to create community infrastructure. The repositories use a common framework in HashiCorp Packer to create repeatable infrastructure. This process allows you to build community applications on AWS with 5 lines of code; to then build and run community application releases.

HashiCorp Packer

USER SUPPORT

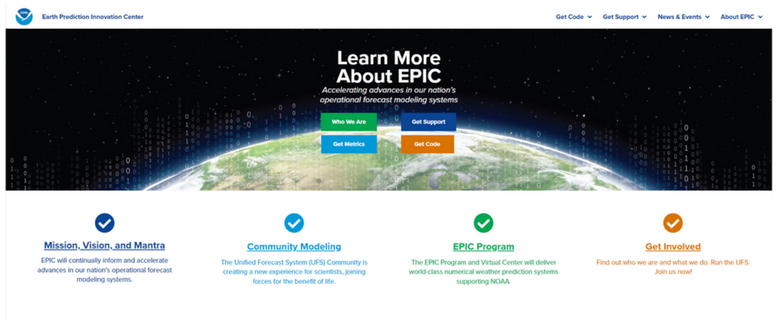
Advanced User Support is a capability that the EPIC community can access, via the EPIC Community Center, for assistance on the identification, improvement, and acceleration of high “Innovation Readiness Level” R2O candidates to UFS Operations.



GitHub Discussions

The technical aspects of running community applications are great. EPIC has a team that supports any questions, comments, idea, or concerns preventing you from conducting research on the supported applications.

CI/CD PIPELINES

Build	Platform Compiler	Unit Test	Static Code Analysis	Run Regression Tests	Package/Pull Artifacts/Deploy
Pull source code from GitHub and stage the data for analysis before deploying the code.	Cheyenne GaeaC5 Gaea Hera Jet Orion Hercules PW-AWS PW-Azure PW-GCP	Run available unit tests for projects and ensure that the tests run as expected. Collect code coverage metrics for the available baselines.	Scan code in all programming languages using SonarQube to determine current vulnerabilities, maintenance issues, and defects.	Run a list of regression tests to test the overall end-to-end functionality.	This gate sequence will package up the artifacts and the application and deploy the application as needed after completing all quality gate checks.

EPIC COMMUNITY PORTAL