XSDK (Extreme-scale Scientific Software Development Kit)

## **Introduction**

The xSDK, or Extreme-Scale Scientific Software Development Kit, is a collection of software components, tools, and policies. It facilitates the development of scientific packages by integrating interconnected software components from various independent groups in the high-performance computing community (“Scientific Software Ecosystems”). To qualify as an xSDK member, a software package must adhere to community policies and demonstrate interoperability with other xSDK libraries (Yang et al.). The initial release in April 2016 showcased the importance of community guidelines in enabling the use and transferability of independently developed software packages. The most recent xSDK release includes 26 mathematical libraries, such as PETSc[1], Trilinos[2], and SuperLU[3], forming a foundation for addressing software interoperability and performance portability challenges. These libraries are continuously evolving to meet the needs of modern scientific computing (“xSDK: Extreme-Scale Scientific Software Development Kit”).

The xSDK envisions a framework that promotes collaboration among diverse teams in the high-performance computing community. By providing interconnected and mutually supportive software components, the xSDK serves as a building block for application development (*xSDK Foundations: Toward an Extreme-scale Scientific Software Development Kit).* Overall, the xSDK aims to establish a software ecosystem that can be easily installed on popular computing platforms. It promotes standardization, documentation, and long-term sustainability while boosting scientific software productivity (“The Exascale Software Portfolio”). Supported by the Department of Energy's Office of Science, the xSDK is designed to initiate a scalable scientific software ecosystem (“xSDK: Extreme-Scale Scientific Software Development Kit”).

## **Key Features**

The xSDK, a scientific computing initiative, distinguishes itself through two key features. Firstly, it adopts a peer-to-peer interoperability approach, utilizing the extensibility of libraries to access functionalities across different packages without the need for additional abstraction layers. This approach facilitates the integration of diverse packages while preserving the existing code structures. Secondly, the xSDK implements software policies that foster compatibility and complementarity among independently developed packages. These policies establish expectations for design, implementation, documentation, support, and installation, allowing separate package development while ensuring their combination and effectiveness. Notably, the xSDK libraries possess capabilities to interact with and modify preexisting data objects that were independently constructed using their own creation procedures (Heroux et al.).

**Interoperability**

The xSDK's interoperability offers a software suite and associated policies that prioritize versatility, reliability, and cost-effectiveness. Through its commitment to promoting interoperability, the xSDK caters to a wide range of user scenarios (“The Exascale Software Portfolio”). At the core of this interoperability is the utilization of Spack[4], an adaptable package manager that facilitates support for multiple versions, configurations, compilers, and platforms. By leveraging Spack, the xSDK automates the determination of dependencies between packages using scripts provided by library developers, resulting in a streamlined building process for complex application codes (Yang et al.). In addition, the xSDK ensures level 1 interoperability across all xSDK libraries by providing a build script, testing environment, and a set of community policies, guaranteeing collaboration and compatibility for users (Curfman McInnes et al.).

**Testing**

The xSDK's testing is currently conducted using Gitlab CI[5], ensuring testing across various platforms. Efforts are underway to enhance the testing capabilities by introducing new tests and additional layers of testing. One such improvement involves testing a subset of xSDK packages to facilitate the identification of potential issues. The main purpose of CI testing is to ensure the stability of the current release version of the xSDK. However, an important secondary benefit is the ability to test with development versions of different libraries, including those with CUDA[6] or OpenMP[7] capabilities, particularly on systems equipped with GPUs[8] or multicore nodes (“Challenges of and Opportunities for a Large Diverse Software Team”).

## **Community Policies**

The xSDK, upon its initial release, made a contribution to the software development landscape by introducing a set of community policies aimed at enhancing the usability, adaptability, and durability of software packages. These policies specifically tackled the hurdles associated with software interoperability and performance portability. One of the key advantages of being xSDK compatible is the integration it offers with other packages, facilitating a cohesive software environment. Moreover, the community installation policies established by xSDK streamline the configuration and installation procedures across diverse platforms, resulting in enhanced efficiency. Through the implementation of these policies, xSDK fosters long-term sustainability and promotes interoperability among packages, forming a foundation for complex applications (Heroux et al.).

## **Scientific Applications**

The xSDK's scientific applications encompass diverse research areas, including surface and subsurface hydrology. Scientists at the Department of Energy utilize software tools like Amanzi[9]/ATS[10], CrunchFlow[11], ParFlow[12], and PFLOTRAN[13], along with xSDK numerical libraries such as hypre[14], PETSc, SuperLU, and Trilinos. They also incorporate Alquimia[15], a specialized xSDK package, to enhance biogeochemical functionality. These efforts contribute to the development of models for hydrological processes.

One prominent application of the xSDK is forecasting the impact of climate change on the Upper Colorado River System. Biogeochemistry models driven by genomic data are integrated with high-resolution hydrology models to understand carbon and nutrient cycles. Notably, stream meanders in the floodplain and the exchange of water between streams and groundwater (hyporheic exchange) significantly affect these cycles. The interconnectedness of groundwater, land surface, and atmospheric feedback is also considered, leading to ongoing studies on ultra-high-resolution integrated hydrology models on regional and continental scales. These models are linked to terrestrial process models and atmospheric models to capture the complexity of the hydrological system (“xSDK: Extreme-Scale Scientific Software Development Kit”).

## **Future Initiatives**

The xSDK project has been expanding its compatibility to include all ECP[16]-funded library efforts, such as SUNDIALS[17] and Magma[18], while addressing performance challenges in exascale computing platforms. Efforts to refine community policies and develop domain component collections tailored to specific communities are underway, enhancing interoperability among scientific libraries. Additionally, xSDK is exploring the use of community installation tools like Spack to ensure reliable installation of its libraries. The project aims to develop interfaces for transfer of computational resources between applications and libraries, optimizing performance in top-level MPI processes (Heroux et al.).

In the future, xSDK plans to adapt its community policies to meet evolving community needs and actively include appropriate packages into its framework. Testing will ensure reliable build and execution, while additional interoperability layers and new example codes will promote collaboration and interoperability between libraries. Specialized xSDK builds with features like CUDA or OpenMP capabilities will be developed to cater to systems equipped with GPUs or multicore nodes. These initiatives will strengthen xSDK's position, drive innovation, and foster collaboration within the scientific computing community (Yang et al.).

**See Also**

* [Exascale Computing Project (ECP)](https://www.exascaleproject.org/)
* [PETSc](https://petsc.org/release/)
* [SUNDIALS](https://computing.llnl.gov/projects/sundials)
* [Magma](http://magma.maths.usyd.edu.au/magma/)
* [Spack](https://spack.io/)
* [CUDA](https://developer.nvidia.com/cuda-zone#:~:text=CUDA%C2%AE%20is%20a%20parallel,harnessing%20the%20power%20of%20GPUs.)
* [OpenMP](https://www.openmp.org/)
* [Scientific computing](https://www.scicomp.uni-kl.de/about/scientific-computing/)
* [Interoperability](https://www.heavy.ai/technical-glossary/interoperability)
* [High-performance computing](https://www.usgs.gov/advanced-research-computing/what-high-performance-computing)

**References**

1. “PETSc 3.19.” *PETSc 3.19 - PETSc 3.19.2 Documentation*, petsc.org/release/. Accessed 2 June 2023.
2. “Trilinos Home Page.” *Trilinos*, trilinos.github.io/. Accessed 2 June 2023.
3. “SuperLU.” *SuperLU: Home Page*, portal.nersc.gov/project/sparse/superlu/. Accessed 2 June 2023.
4. *Spack*, spack.io/. Accessed 2 June 2023.
5. “Gitlab CI/CD.” *GitLab*, docs.gitlab.com/ee/ci/. Accessed 2 June 2023.
6. “Cuda Zone.” *NVIDIA Developer*, 13 Oct. 2022, developer.nvidia.com/cuda-zone#:~:text=CUDA®%20is%20a%20parallel,harnessing%20the%20power%20of%20GPUs. Accessed 2 June 2023.
7. “Home-OpenMP.” *OpenMP*, 14 Mar. 2023, [www.openmp.org/](http://www.openmp.org/). Accessed 2 June 2023.
8. “Graphics Processor Unit (GPU): What, Type, How & Definition.” *Teach Computer Science*, 21 Dec. 2021, teachcomputerscience.com/graphics-processor-unit/. Accessed 2 June 2023.
9. Laboratory, Los Alamos National. “Scientific Software.” *High-Performance Computational Methods and Software Tools in Support of Environmental Health, Cleaner Energy, and National Security.*, organizations.lanl.gov/cels/ees-division/scientific-software/. Accessed 2 June 2023.
10. Xu, Zexuan, Molins, Sergi, Dwivedi, Dipankar, Özgen-Xian, Ilhan, and Steefel, Carl. *ATS (Advanced Terrestrial Simulator) integrated hydrology and reactive transport model output in Copper Creek, Colorado*. United States: N. p., 2022. Web. doi:10.15485/1877377.
11. “Crunchflow.” *IDEAS*, ideas-watersheds.github.io/software-ecosystem/codes/crunchflow/. Accessed 2 June 2023.
12. *ParFlow Hydrologic Model*, 7 July 2020, parflow.org/. Accessed 2 June 2023.
13. *PFLOTRAN*, www.pflotran.org/. Accessed 2 June 2023.
14. “Hypre: Scalable Linear Solvers and Multigrid Methods.” *LLNL Computing*, computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods. Accessed 2 June 2023.
15. Andre, Benjamin, Molins, Sergi, Johnson, Jeffrey, Steefel, Carl, and USDOE. *Alquimia*. Computer software. *https://www.osti.gov//servlets/purl/1231758*. USDOE. 1 Aug. 2013. Web. doi:10.11578/dc.20210416.49.
16. “Home Page - Exascale Computing Project.” *Exascale Computing Project*, www.exascaleproject.org/. Accessed 2 June 2023.
17. “Sundials: Suite of Nonlinear and Differential/Algebraic Equation Solvers.” *LLNL Computing*, computing.llnl.gov/projects/sundials. Accessed 2 June 2023.
18. “Magma Computational Algebra System.” *Magma*, magma.maths.usyd.edu.au/magma/. Accessed 2 June 2023.

**Bibliography**

Curfman McInnes, Lois, et al. “What Are Interoperable Software Libraries? - Ideas-Productivity.Org.” *IDEAS Productivity*, 5 Dec. 2016, ideas-productivity.org/wordpress/wp-content/uploads/2016/12/IDEAS-InteroperabilityWhatAreInteroperableSoftwareLibraries-V0.2.pdf.

Heroux, Michael A., Bartlett, Roscoe, Demeshko, Irina, Gamblin, Todd, Heroux, Michael A., Johnson, Jeffrey, Klinvex, Alicia Marie, Li, Xiaoye, McInnes, Lois Curfman, Moulton, J. David, Osni-Kuffuor, Daniel, Sarich, Jason, Smith, Barry, Willenbring, James Michael, Yang, Ulrike Meier, and Hammond, Glenn Edward. *xSDK Foundations: Toward an Extreme-scale Scientific Software Development Kit*. United States: N. p., 2017. Web. doi:10.14529/jsfi170104.

Heroux, Michael A. “Scientific Software Ecosystems.” *CSBSJU*, faculty.csbsju.edu/mheroux/SciSWEcosystems.pdf. Accessed 22 May 2023.

Noecker, Curtis. “Making Scientifific Applications Portable: Software Containers and Package Managers.” *DigitalCommons@CSB/SJU*, 2018, digitalcommons.csbsju.edu/cgi/viewcontent.cgi?article=1045&context=honors\_thesis. Accessed 22 May 2023.

“The Exascale Software Portfolio.” *Science and Technology Review*, str.llnl.gov/2021-02/diachin. Accessed 22 May 2023.

“xSDK: Extreme-Scale Scientific Software Development Kit.” *xSDK*, xsdk.info/. Accessed 22 May 2023.

Yang, Ulrike Meier, and McInnes, Lois Curfman. *xSDK: Building an ecosystem of highly efficient math libraries for exascale*. United States: N. p., 2021. Web. Accessed 22 May 2023.

Yang, U. M., et al. “Challenges of and Opportunities for a Large Diverse Software Team.” *GitHub*, collegeville.github.io/CW21/WorkshopResources/WhitePapers/challenges-of-and-opportunities-for-a-large-diverse-software-team.pdf. Accessed 22 May 2023.