

Development of SSC-NG

the Next Generation SSC Archive

Christopher F Baum, Lars Vilhuber

Stata Conference, October 2026



SSC-NG

Statistical Software Components — Next Generation

[About](#)

[Development Plan](#)

[Governance](#)

[Funding](#)



SSC-NG

Modernizing the Statistical Software Components (SSC) archive for Stata — a robust, community-governed, next-generation infrastructure.



About SSC-NG

Learn about the project, its governance, and initial funding.

[Read More](#)



SSC Archive by Date

Browse and install packages from the community-run time-stamped archive of the SSC.

[Visit Archive](#)



Legacy Submission Site

Submit a package today using the legacy SSC submission process.

[Submit a Package](#)

`ssc describe ssc2, ssc install ssc2` provide access to the 'snapshot' version of the SSC Archive from December 2021 to the present day.

A 3-year project supported by the Alfred P. Sloan Foundation, Lars Vilhuber, coordinator

Tasks:

- 1 Coordination and Governance by a Preliminary Governing Council
- 2 Design SSC-NG
- 3 Solicit community Input
- 4 Enhance the SSC-NG repository
- 5 Establish persistent non-profit support structure

While Stata is a commercial product, the SSC archive is entirely community-run, relying on software created and contributed by volunteers from multiple disciplines. SSC has successfully fostered community contributions for decades, but its current infrastructure has personnel and organizational fragility, limits the findability and accessibility of its resources by modern standards, and is maintained through primarily manual processes.

This project — **SSC Next Generation (SSC-NG)** — will develop the tools, organizational structure, and infrastructure needed to address these issues and prepare the SSC archive for the next decade, while preserving strong backward compatibility with the present system. We do not intend to break what works well, but to make the parts that need to be more robust, and more efficient.

Task 2: Design SSC-NG

Design a system that supports both email submission (for legacy users) and a web-based submission system that enables automation, with authentication of contributors and protection against spam or spurious submissions. Key elements planned for the first year:

- **General web hosts and email addresses** not tied to a single person or institution, redirecting transparently to existing mechanisms at first, then transitioning to new ones.
- **Automated checks and tests**, starting with simple checks (e.g., no collisions with existing function names), run across multiple Stata versions, with unit tests and sample run code added over time.
- **arXiv-style moderation**, where volunteer subject-matter specialists approve in-scope submissions without vetting utility or correctness.
- **Julia-style register**, using pull requests against a central, GitHub-hosted catalog to build the physical SSC package repository, while remaining backward-compatible with current installation methods.

- **Interim infrastructure:** centrally hosted Git repositories within a new GitHub organization for authors without their own repository, wound down later for actively maintained packages but preserved for unmaintained ones.
- **Fully redundant hosting:** because Stata packages are lightweight, a complete copy of the SSC catalog can be taken into physical custody rather than relying on links to author-controlled repositories that may disappear.
- **Improved preservation:** regular snapshots (e.g., monthly full and daily incremental) pushed to a trusted archive with an API, such as Zenodo or a Dataverse instance, with tooling to “re-hydrate” the archive.
- **Date-based preservation,** similar to CRAN’s archival function and the date-based functionality of the (now defunct) MRAN / the current Posit Package Manager.
- **Compliance with existing infrastructure:** continued indexing in RePEc, continued functioning of `ssccount` and `ssc hot` , and a smooth transition worked out with Stata, Boston College, and RePEc maintainers.

Task 4: Enhance the SSC-NG Repository

Building on Task 2 and informed by Task 3, make the submission system more scalable, efficient, and automated:

- **Streamlined submissions:** a website for developers without a GitHub- or GitLab-based repository, and a submission bot for those who have one.
- **Reputation-based automatic acceptance / sponsored initial acceptance,** tested and implemented along the lines of the Julia registry, informed by approaches such as pyOpenSci and rOpenSci.
- **Version-based preservation,** adding checks that warn submitters when metadata such as a version number is absent or inconsistent.
- **Tutorials for developers:** a package template with step-by-step guidance on creating, testing, and submitting packages, with an emphasis on automated submission.

- **Central documentation for all packages:** parsing the full SSC archive and generating consistent HTML help pages, incorporating richer author-maintained documentation where it exists.
- **Improved citability of packages:** a new metadata standard for self-citation of Stata packages, and a Stata package that makes it easy to cite other packages.

Task 5: Persistent Non-Profit Support Structure

In parallel with Task 4, define and implement a persistent non-profit support structure, along one of two paths:

- A 501(c)(3)-compliant charitable organization as a formal LLC or corporation, or
- Association with a fiscal sponsor that is itself a 501(c)(3)-compliant non-profit, such as Software in the Public Interest (SPI) or Open Source Collective (OSC).

In either case, the PGC's role is to define bylaws and a business model, and to hand over a functioning entity to the successor non-profit organization.