

**Response to the NASA Public Access Plan for Increasing Access to the Results of
NASA-Supported Research**

Dr. Rebecca Ringuette, NASA Goddard Heliophysics Digital Resource Library

Dr. Brian Thomas, NASA Goddard Heliophysics Digital Resource Library

Dr. Robert Candey, NASA Goddard Space Physics Data Facility

Designing a Efficient Software Infrastructure for Validation

We applaud the Public Access Plan for including NASA-funded software development as a vital research artifact to be preserved. However, we ask that the stance be firmer, specifically to *require* software archival instead of *recommending* archival. Changing this stance to be firmer requires some changes in policy and infrastructure: (1) a relaxation of NASA policies to claim ownership of hosted softwares, (2) an augmentation to the current archiving infrastructure, and (3) additional factors to be considered in SMPs and DMPs. We are persuaded that the Public Access Plan's stated goal of publication validation is impossible without strengthening this stance.

Recent efforts predating the newly revised NPR 2210 to archive software with a domain-specific repository failed due to the NTR release process. Despite efforts to host a modeling code at a NASA repository, the Heliophysics IRI modeling code had to be hosted separately from its proper domain-specific repository (HDRL/SPDF) since some of the developers were external to NASA (<https://irimodel.org/>). This example is one of many across the science archives and speaks to a software archiving system that is broken at the root. This should not be the case. In addition to the recent changes in NPR 2210, **we ask that the NASA software ownership requirements be relaxed so that software developed in collaboration with those outside of NASA can be archived at a NASA domain-specific repository without NASA claiming complete ownership of the code.**

Leo Singer of the NASA Goddard Open Science Team has researched this problem in more detail, along with related problems on software licensing barriers, and is available for comment upon request. The end result of the changes we are requesting is for the research community to more easily archive software associated with a publication next to the related datasets at a domain-specific archive without a complex licensing and rights negotiation, which currently fails too often. Yes, NASA should be recognized for the software it produces, but it should also make an easy pathway to host software developed by and in collaboration with others at the appropriate repository for the domain *without* signing away their rights to the software. NASA cannot provide long-term research artifact preservation for the publications it funds if it cannot work out a way to host these artifacts as other entities have done. Artifacts hosted elsewhere will simply be lost, further burdening our researchers in their efforts to build on previous work.

Turning to the second change, **we advocate for a software infrastructure to be created that builds upon the current data infrastructure** instead of building a more generic software archiving infrastructure from the ground up. Including software curation and archiving into the currently existing data curation and archiving infrastructure has several advantages over a more generic structure. Data curation scientists at domain-specific archives already understand the curation process necessary for their specific domain, and in many cases are already familiar with some basic curation processes for software that accompanies datasets (e.g. mission processing software). They are the best choice to train software curation scientists on the domain-specific curation methods, likely hired from the currently-existing software peer-review and development community that already understands usability and preservation issues, such as the Journal of Open Source Software (JOSS), pyOpenSci, ROpenSci, and PyHC. Additionally, a domain-specific software archiving system can easily adapt their search

interfaces to include software, usually by simply adding a ‘software’ tab to the search interface as done for the NASA Science Discovery Engine, resulting in a more powerful search interface. The search capabilities of the domain-specific repositories can be linked to such efforts as is done for the data.nasa.gov website. We note that the depth of curation necessary for software supporting a publication is generally less than that required by JOSS and pyOpenSci, but will vary based on the complexity of the analysis in the publication.

Another benefit to an augmentation approach is the datasets’ and softwares’ metadata can be more easily re-curated to link to newly related artifacts (e.g. new publications and newly associated datasets) if they are stored at the same location and managed by the same curation team. Building a generic software archival system separate from the current infrastructure has none of these benefits and promises to take a much longer amount of time and more funds to develop. As an efficient software infrastructure is created, we expect the software metadata records should be made searchable on the software.nasa.gov website, just as the dataset metadata records are to be searchable on the data.nasa.gov website. Suggestions on how to link resources of differing types into a search interface are given in <https://doi.org/10.1016/j.asr.2022.10.051>.

One possible pathway for domain-specific repositories to perform software curation is as follows. Authors can upload their analysis scripts and software to a private repo on the generalist code repository website most relevant for their domain as instructed by the domain-specific data and software repository. The author can then share a link to that repo with the software curation scientist, who will then provide feedback on the curation tasks appropriate for that software. Once those curation tasks are completed, the curator can then “fork” the software repo into the domain-specific repository, incorporate that artifact into the search mechanisms of that repository, and assume responsibility for preserving those scripts and updating its metadata over time. The DOI will be assigned to this fork of the software. This example includes the use of a generalist repository such as GitHub, but **places the final preservation, searching, access, and re-curation responsibility on the domain-specific archive**, as it should be.

The goal of software curation should at minimum be a landing page similar to a dataset providing basic information. That landing page should describe the software and what it does, list all supporting packages needed, indicate the programming language and version, guide the reader on how to run the scripts, indicate what portions of the scripts are related to what sections of the publication, and include licensing information, access restrictions, and any other information likely to be needed in validation efforts. Notably, software curation does not need to include preservation of the software environment, but must at least provide a record of what was done to produce the result in a given publication (e.g. run script A on dataset AB then run script 2 on the output of script A, etc). An improvement upon this that should be considered optional is for softwares to reference containerized software environments in which the software is known to run successfully in. Such an array of containerized environments is in development in Heliophysics, so we reserve requesting that this be required for a future date (e.g. <https://hub.docker.com/u/spolson>). We ask that **Software Management Plans (SMPs) be**

required to include plans for such metadata and be reviewed by the destination archive in proposal review processes during proposal selection and after project completion.

We agree with the comment (Part C Section 1.0) that openness should be balanced with level of effort, and we point this out as an **opportunity for proposers to gain open science ‘points’** in their proposals. For example, we would expect a proposal including more openness in their final product to be preferred over one that doesn't, all other considerations being approximately equal. Providing a detailed list of possible activities that proposers can include in the DMPs and SMPs to create more open data and software products would be useful to proposers, and should be accompanied by direction in the AO for the proposers to choose actions appropriate for their proposed work, and justify their choices based on what is expected to be necessary for validation efforts and any choices expected to add value to those products for the community. Activities not included in the list should be considered, and possibly added to the list of possible activities in updates to the AO. These options allow proposers a way to satisfy some minimum requirement appropriate for their category of work as determined by NASA and the destination repository, but also provide the proposers flexibility to increase a project's openness by choosing tasks that are reasonable for them based on their skills, resources, and relevance to that project. For example, one proposer might choose to build software installation mechanisms into their final product, while another may develop detailed examples and executable notebooks. The usefulness of a given task will vary from project to project, and should be determined jointly by the proposal reviewers and destination archive. Here we are also implicitly asking for collaboration between NASA and the domain-specific repositories on what is appropriate, which could be simply requiring the proposer to discuss their SMPs and DMPs with the selected repository and obtain initial approval before submitting. Ideally, the full range of these options would originate from community brainstorming workshops or other input methods that gather such suggestions from the science and software development communities.

Beyond these two major points, we also point out **some important additions and changes** we request be made to the NASA Public Access Plan. Requiring DOIs for datasets and not for software decreases the perceived value of software compared to datasets, despite NASA's efforts to describe otherwise in the plan. This differing treatment also makes software more difficult to reference and to link to related publications and datasets. We request that DOIs also be required for software developed with NASA funds. Also, **software related workforce development** should be mentioned alongside related dataset training in the policy, ranging from simple topics such as 'How to create a useful readme file' to more complex needs, such as a summer school guiding users on how to install and use modeling codes in cloud environments. Training on best practices for both software and data are greatly needed by the research community. We comment that best practices are in various stages of development and have yet to be developed into guidelines or rubrics for many of our science communities, such as what datasets and scripts should be included to properly support peer-review validation for a given publication type (e.g. <https://modeldatarcn.github.io/>). Such research should be performed by the community and supported by AOs, possibly through the currently existing ROSES open science AOs.

Part A Section 2.0 uses a definition for research data that includes all material “necessary to validate research findings” (top of page 7). Since publications, analysis scripts and related software are necessary to validate research findings, this definition of ‘data’ includes software and publications, yet the word ‘data’ is used in the same section of the policy to refer only to datasets and Parts B and C address publications and software. Please add a statement pointing out that while the definition includes datasets, publications, and software as research artifacts necessary to validate research findings, the policy addresses these items in separate sections. Such a statement will reduce confusion in understanding the policy.

We ask that the sections on **compliance processes and metrics** include comments that development of those processes will be done with **community input through workshops** and RFIs as deemed appropriate. We are already planning a series of workshops for Heliophysics on those topics. We have communicated this effort to some at NASA HQ and have received positive feedback. We are happy to collaborate with other science directorates and agencies on those efforts to make them more useful and adaptable to those entities’ efforts. Also, Part B does not have a section 4.

Please change Part B section 5.0: “Publication metadata made available in parallel with final acceptance of the paper, coded for machine readability, and available without charge. A link to the publication and any supplemental materials **must** also be provided,” where the bold word is the requested change (from ‘will’ to ‘must’). Publication metadata is not fully useful without a link to the publication and *all* supplemental materials, which are preferably hosted at a domain-specific repository.

We also ask that published articles be required to include a link to the related software or metadata landing page, which **must** indicate any restrictions on access to that software (the A in FAIR). We ask that software documentation be required to include how the software was installed. This is vitally important for modeling codes, and lack of such instruction is preventing portions of the community from reusing those codes in Heliophysics. We ask the policy to encourage the most open license possible be selected for the software, and that portions of restricted softwares without restricted information be made open to the community (e.g. a partial or ‘redacted’ release of the software). SMPs should also include the expected license choice based on conversations with all parties, and justification for any expected access and usage restrictions.

Finally, we ask that infrastructure be funded to collaborate together and with commercial entities to streamline various required activities, such as metadata curation and search interfaces. The NASA Data Repositories Workshops are a great beginning, but we also need ‘hackathon’ style workshops where we can sit down with someone who built something we are interested in adopting and work out how to build it into our services. It would be additionally beneficial to pay commercial entities to assist in related efforts to accelerate improvements to our infrastructure.

We have submitted a separate RFI addressing current growing barriers in infrastructure for validation. Thank you for your attention.