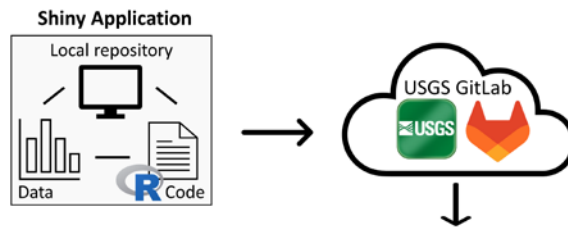


Ecosystems Mission Area—Land Management Research Program

Best Practices for Distributing and Deploying U.S. Geological Survey Shiny Applications

Chapter 2 of
Section B, Web Applications, of
Book 7, Automated Data Processing and Computations



Techniques and Methods 7–B2

Cover. An example of a USGS Shiny application displaying the required visual elements for public distribution. Input menus and an interactive web map allow users to interact with the underlying scientific data. This application is hosted through USGS Cloud Hosting Solutions at https://rconnect.usgs.gov/Climate_Distance Mapper/. Following procedures described in this report, the application was paired with an interpretive USGS product (journal article; <https://doi.org/10.1002/ecs2.2453>) and a USGS data release (<https://doi.org/10.5066/P9FCRGHF>). As shown in the accompanying diagram, USGS Shiny applications can be developed in a local repository and released through USGS OpenSource GitLab. Image by Daniel F. Shryock, U.S. Geological Survey, August 1, 2023.

Best Practices for Distributing and Deploying U.S. Geological Survey Shiny Applications

By Daniel F. Shryock, Micah C. Wright, Phillip J. van Mantgem, and Todd C. Esque

Chapter 2 of
Section B, Web Applications, of
Book 7, Automated Data Processing and Computations

Ecosystems Mission Area—Land Management Research Program

Techniques and Methods 7–B2

U.S. Department of the Interior
U.S. Geological Survey

U.S. Geological Survey, Reston, Virginia: 2023

For more information on the USGS—the Federal source for science about the Earth, its natural and living resources, natural hazards, and the environment—visit <https://www.usgs.gov> or call 1–888–392–8545.

For an overview of USGS information products, including maps, imagery, and publications, visit <https://store.usgs.gov/> or contact the store at 1–888–275–8747.

Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Although this information product, for the most part, is in the public domain, it also may contain copyrighted materials as noted in the text. Permission to reproduce copyrighted items must be secured from the copyright owner.

Suggested citation:

Shryock, D.F., Wright, M.C., van Mantgem, P.J., and Esque, T.C., 2023, Best practices for distributing and deploying U.S. Geological Survey Shiny applications: U.S. Geological Survey Techniques and Methods, book 7, chap. B2, 10 p., <https://doi.org/10.3133/tm7B2>.

ISSN 2328-7055 (online)

Contents

Abstract.....	1
Overview of U.S. Geological Survey Policy in Relation to Shiny Applications.....	1
Required Reviews for Scientific Software	3
Provisional Software Release	3
Software Release Process for Shiny Applications	4
Brief Glossary of Git Terminology Relevant for Software Releases.....	4
General Recommendations.....	5
Recommended Workflow for Scientific Software Release	5
Data Releases Associated with Shiny Applications.....	6
Shiny Application Requirements and Best Practices.....	6
Standards for Visual Appearance.....	6
Disclaimers	7
Deployment of Shiny Applications to the Web	8
Testing.....	8
Error Messages.....	8
Documentation.....	8
Versioned Updates to Shiny Applications	8
Summary.....	9
Acknowledgments	9
Glossary.....	10

Figures

1. Flowchart showing U.S. Geological Survey review and approval process for Shiny applications
2. Image showing USGS appearance requirements for web pages.....

Abbreviations

CHS	Cloud Hosting Solutions
FSP	Fundamental Science Practices
HTML	hypertext markup language
IPDS	Information Products Data System
USGS	U.S. Geological Survey

Best Practices for Distributing and Deploying U.S. Geological Survey Shiny Applications

By Daniel F. Shryock, Micah C. Wright, Phillip J. van Mantgem, and Todd C. Esque

Abstract

A goal of U.S. Geological Survey (USGS) research is to create actionable science for U.S. Department of the Interior managers and partners. Properly managed and released software tools can quickly, accurately, and easily translate scientific results to aid managers' decision-making. The USGS policies on software development and distribution affect how authors of USGS software products must navigate the review and approval process. Understanding these rules and procedures will save time and effort in making science useful and visible through software development. This document outlines USGS policies on software development as they pertain to Shiny applications, which use R code to create interactive web-based applications. The requirements and procedures described here will also be relevant for other USGS software applications.

This document is not intended to be a step-by-step guide on building a Shiny application or to go into detail about any of the referenced software programs. An understanding of the referenced software programs, including R, Git, and hypertext markup language/cascading style sheets (HTML/CSS) is needed to use this document efficiently, though links for more thorough guidance are included throughout.

Overview of U.S. Geological Survey Policy in Relation to Shiny Applications

The review process outlined in this document applies to all U.S. Geological Survey (USGS) software releases. However, because this document is designed to help USGS scientists deploy Shiny applications, it provides additional

information where necessary regarding Shiny application development. Shiny applications are a popular platform using R code to create interactive web applications that execute a series of R functions and can be designed to generate user output, allow interaction with scientific data or algorithms, and visualize results. Shiny applications can be deployed on local workstations when distributed as a package of source code and data files or over the web when deployed on a public-facing server. Review as a USGS software product is required for public deployment and distribution.

Per USGS policy (Instructional Memorandum [IM] OSQI 2019-01), software is defined as follows:

“(1) computer programs that comprise a series of instructions, rules, routines, or statements, regardless of the media in which recorded, that allow or cause a computer to perform a specific operation or series of operations; and (2) recorded information comprising source code listings, design details, algorithms, processes, flow charts, formulas, and related material that would enable the computer program to be produced, created, or compiled.”

For further information, refer to the USGS software management website (<https://www.usgs.gov/products/software/software-management>). Additional information on specific topics can be found in the frequently asked questions for software development (<https://www.usgs.gov/office-of-science-quality-and-integrity/e6-software>).

The USGS has additional policy guidance for Scientific Software, defined as follows:

“Software containing source code implementing scientific algorithms or producing scientific data. It may include any custom developed code yielding scientific results, thereby facilitating a clear scientific workflow of analysis, scientific integrity, and reproducibility.”

2 Best Practices for Distributing and Deploying USGS Shiny Applications

Scientific Software can be a stand-alone USGS product or associated with other products, such as USGS data releases, open-file reports, or journal articles. Most often, Shiny applications will implement scientific algorithms or produce scientific data and therefore need to be considered Scientific Software. For this reason, Shiny applications will generally be subject to the USGS review and release policy described in IM OSQI 2019-01. Scientific Software is tracked through the Information Product Data System (IPDS) and must obtain Science Center Director approval before release as an approved USGS information product. Software applications, including Scientific Software, are classified as non-interpretive products in the USGS Fundamental Science Practices (FSP)

policy (see <https://www.usgs.gov/about/organization/science-support/office-science-quality-and-integrity/levels-delegated-bureau>). Consequently, Shiny applications that contain new interpretive information (see <https://www.usgs.gov/office-of-science-quality-and-integrity/distinctions-between-new-research-or-interpretive>) must also be paired with a USGS publication series or journal article because these are the only information products allowed to advance new interpretive information (fig. 1). Shiny applications can be deployed on the web for public access once an authoritative version of the source code has been approved for release, and all associated information products (for example, data releases) have been approved.

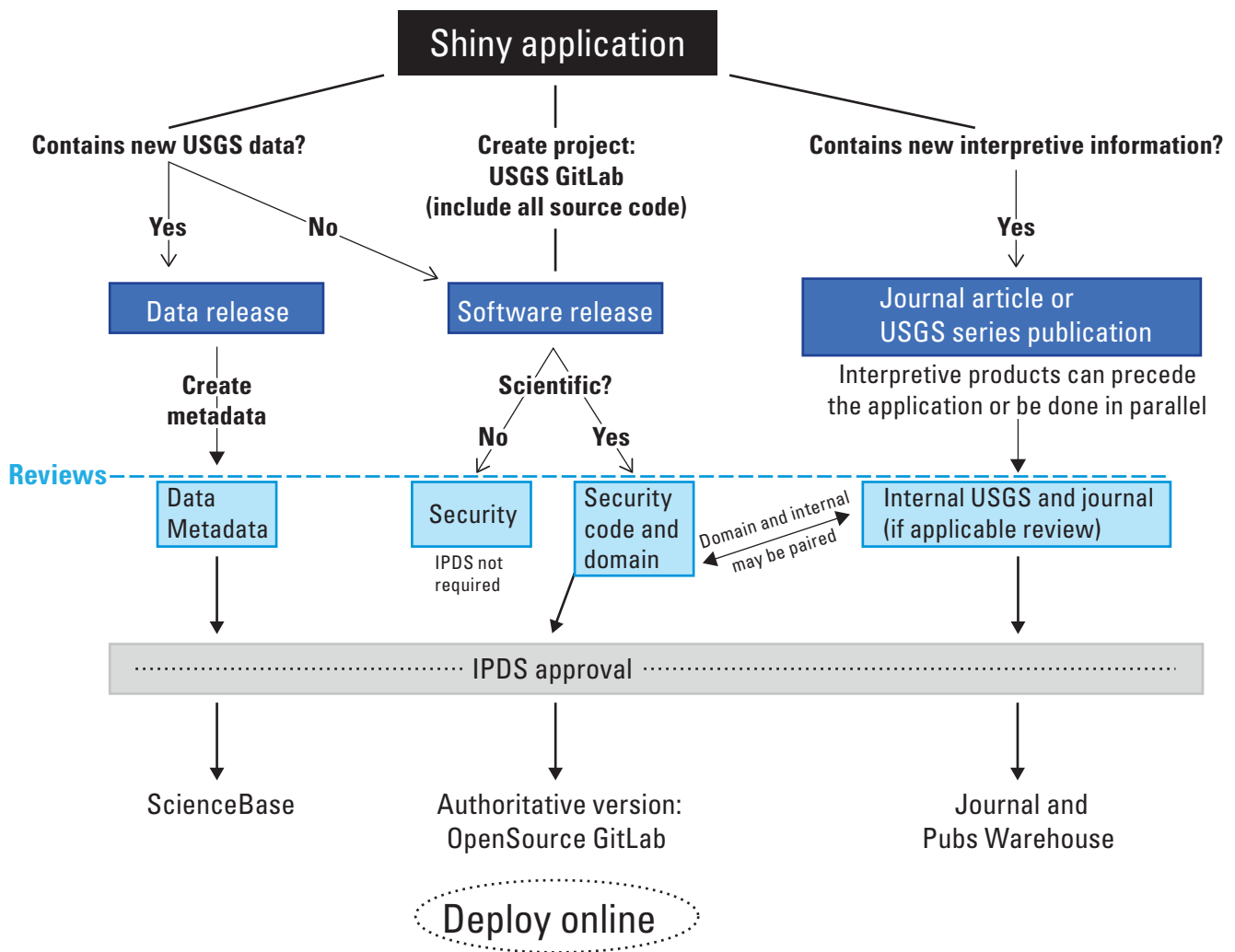


Figure 1. U.S. Geological Survey (USGS) review and approval process for Shiny applications. Not all steps are required, and many depend on the nature of the application. IPDS, Information Products Data System.

Authors of Shiny applications will need to work with their chain of command (for example, the Principal Investigator [PI], Research Manager, and Science Center Director) to determine whether a Shiny application rises to the level of Scientific Software. When Shiny applications are deployed as “websites that are created as part of a communication strategy for sharing scientific results that are published and made available elsewhere” (see <https://www.usgs.gov/products/software/software-management>), they may be treated as non-Scientific Software that are only subject to administrative security review (code and domain reviews are still recommended, but not required; see the software frequently asked questions linked earlier in the text). In non-Scientific software, scientific results must have been previously authorized as part of an associated interpretive IPDS product (for example, a journal article or other USGS series publication). Non-Scientific Shiny applications must not contain new interpretive information, as defined in the FSP (see “Distinctions Between New Research or Interpretive Information Products, Previously Published or Noninterpretive Information Products, and Scientific Data” at the hyperlink mentioned earlier in the text) and must not extrapolate or predict from scientific algorithms. Only interactive data visualizations such as graphs, maps, or tables of previously published interpretive data are acceptable for non-Scientific Software deployed to the web. Hence, this classification is analogous to the requirements for a USGS web-page product (see “Levels of Delegated Bureau Approval Authority for U.S. Geological Survey Information Products” at the hyperlink mentioned earlier in the text).

Required Reviews for Scientific Software

All USGS Scientific Software must be tracked via IPDS. Approval to publish Scientific Software is contingent on three types of reviews: (1) administrative security review, (2) code reviews, and (3) domain review (<https://www.usgs.gov/products/software/software-management/types-software-review>). Shiny applications (or similar products) classified as Scientific Software must undergo this review process for approval as an official information product. These reviews serve to prevent the distribution of personal or sensitive information, to ensure code quality, accuracy, and efficiency, and to confirm that algorithms match those in the published literature. When a reviewer has sufficient expertise and is willing, the same reviewer may perform multiple reviews. The three types of review are briefly defined:

- (1) Administrative security review ensures that the source code and any associated files do not contain personal, private, or otherwise sensitive information (for example, personally identifiable information, absolute file paths, internal Internet Protocol [IP] addresses, or usernames/passwords). This review can be performed by any trusted individual.

- (2) Code review ensures structural code quality and can be performed by an individual with sufficient programming knowledge. Even if specific coding standards are not specified, the USGS software development page (hyperlink mentioned earlier in the text) suggests that reviewers look for proper formatting, code efficiency/optimization, security vulnerabilities, and unit testing.
- (3) Domain review ensures that software produces results that align with all applicable published science products. For example, the software should produce results consistent with a previously published journal article describing an algorithm. This review can be performed by a scientific subject matter expert.

Provisional Software Release

When extensive cooperation with external collaborators is necessary to develop a Shiny application, approval may be granted to release provisional software before formal approval as a USGS scientific information product. The software must then contain the appropriate disclaimers for provisional releases (see “Disclaimers” and “FSP” at <https://www.usgs.gov/about/organization/science-support/office-science-quality-and-integrity/fundamental-science-5>) and must not contain personal, private, or otherwise sensitive information. Administrative security reviews are still required. IPDS tracking is not required at the provisional stage, but software must receive Science Center Director approval before public release. Although no time limits are given concerning the lifespan of provisional releases, any authoritative versions must go through the full release process, including all reviews, documentation, and approvals.

Detailed guidance on review requirements is available on the software management page (<https://www.usgs.gov/products/software/software-management/additional-guidance-review-and-approval-scientific-software>). In particular, the following section is relevant to sharing with outside collaborators:

“All Scientific Software needs formal Center Director approval documented in IPDS. Peer reviews must be completed in IPDS prior to submission to Center Directors. Provisional or preliminary software is subject to revision and often released prior to formal approval as a USGS scientific information product in order to support collaborative software development with colleagues and partners from outside entities, however it must still be approved by a Science Center Director before any type of public release occurs.”

Software Release Process for Shiny Applications

The following procedure is applicable for Shiny applications classified as Scientific Software. The USGS has one approved repository for official software product releases, USGS OpenSource GitLab (<https://code.usgs.gov>). For this reason, the files for any Shiny application intended for public release (including provisional releases) will need to be uploaded into a USGS OpenSource GitLab project to undergo the review and release process. During the software development process, USGS employees may request external collaborator accounts on this platform. Before release, projects may be kept as internal or private with external collaborators. A second code repository is available to USGS employees—USGS InnerSource GitLab—with much of the same functionality. However, this site is only available to authorized users on the U.S. Department of the Interior network. To facilitate public release, software projects developed on InnerSource GitLab will need to be migrated to USGS OpenSource GitLab before final approval.

Git is a widely used version control software and is the only version control system supported by the USGS (see <https://git-scm.com/about> for more details). The USGS OpenSource GitLab is a web-based Git repository manager hosted by USGS Cloud Hosting Solutions (CHS). The USGS Software Development page (see hyperlink mentioned earlier in the text) provides a useful comparison chart (<https://www.usgs.gov/media/images/image-usgs-git-comparison-table>) for details on the different Git interfaces available to USGS employees. The three required reviews (administrative security, code, and domain reviews) for Scientific Software can be documented through the GitLab issue tracking feature. A brief description of Git and GitLab is given in the following text, along with some helpful links for more detailed information.

Brief Glossary of Git Terminology Relevant for Software Releases

The workflow presented in the following section requires Git and USGS GitLab and assumes some familiarity with Git terminology. Although a comprehensive overview of Git is beyond the scope of this document, we provide a brief glossary and a few helpful links.

- What is Git?
 - Git is a version control software for source code. A useful analogy is track changes in Microsoft Word, but with more flexibility and better record keeping. Git allows for application development without risk of losing previous versions, though developers must decide when to commit changes for tracking.

- What is GitLab?
 - GitLab is a web-based Git-repository manager. Although Git can be used without a repository manager (GitLab or others like GitHub), it is required for the USGS software release process.
- A brief glossary of key Git terminology.
 - Commit: a snapshot of the source code file or files at a specific time. The term “commit” is used as a noun (the snapshot) and a verb (the act of “taking” the snapshot).
 - Branch: the version lineage. Git can track multiple branches for the same project, which allows developers to work on updates to a project without changing the core, stable branch. This branch is often called the master branch. The term “branch” can also be a verb (the act of creating a branch) and a noun (the branch itself). Commits belong to a specific branch.
 - Push: upload Git history (new commits) to a remote repository.
 - Remote: short for remote repository; this is what is stored on GitLab, but technically a remote repository is just another version of the repository. Often, the remote repository is not on the developer’s local computer. Changes made to the developer’s local repository on their personal workstation can be pushed to the remote repository at any time.
 - Repository: a database containing a project’s history. The commits are stored here.
 - Tag: a reference pointer for a specific object, usually a commit.
- Full glossary of Git commands and terminology (<https://git-scm.com/docs/gitglossary>).
- Git documentation (<https://git-scm.com/docs>).
- Version control in RStudio (<https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN>).
- Getting started with GitLab (<https://docs.gitlab.com/ee/intro/>).

General Recommendations

- Identify reviewers for all three types of software review at the beginning of the project. This identification helps avoid unnecessary delays and allows reviewers time to complete the review process in a timely manner.
- If the application is being released with an associated paper, consider having the internal paper review and software domain review completed in tandem.
- If the application is associated with a USGS data release, begin the process of data review and release as early as possible.
- All committed code in a repository history must receive an administrative security review. Initialize security reviews at the beginning of the project so changes can be incrementally reviewed. The code and domain reviews can be performed once before project release, unless substantial changes are made that require additional reviews (to be determined by the Science Center Director).
- Test the software as much as possible before initiating reviews to avoid any unnecessary back and forth. Run the application with as many different computers and users as possible to discover unexpected bugs, incomplete documentation, and other issues.
- Use the issues feature in Git to record reviews and review response. If any changes are recommended, create a document listing those changes and author response.
- When releasing software, create a separate release candidate branch.

Recommended Workflow for Scientific Software Release

1. Begin the data review process if the application contains new USGS data and is associated with a USGS data release.
2. Develop the application as an internal or private project in USGS OpenSource GitLab.
3. Designate the software reviewers for the administrative security, domain, and code reviews. This should be done as early in the project timeline as possible to allow reviewers time to complete reviews and avoid unnecessary delays.
4. Perform administrative security reviews as early and often as possible. When a GitLab project is made public, the whole project can be viewed, including all commits in any branch that have been pushed to GitLab. Therefore, the administrative security review must cover the entire history that is pushed to GitLab (every commit in every branch), which can be arduous if done all at once.
5. Create a project on GitLab and push the local repository (see <https://docs.gitlab.com/ee/user/project/repository/index.html>). Make sure that the project is in an appropriate namespace for your organization. For example, projects for the Western Ecological Research Center can be found at <https://code.usgs.gov/werc>. You may need to request access to create a new project in the appropriate namespace. At this point, the project visibility must be set to either internal or private.
6. All USGS GitLab projects must have the following items: code.json metadata file (template: <https://www.usgs.gov/products/software/software-management/distribution-usgs-code>), license file (see previous link), and disclaimer file. A readme file describing the project is a good addition as well. The license, readme, and disclaimer files can be created as markdown (.md) files that will render as HTML when viewed on GitLab. They do not need to be rendered to HTML beforehand, and the HTML version does not need to be tracked in Git. Rmarkdown files (.Rmd) will not automatically render as HTML in GitLab. At this point in the project, disclaimers should use the “provisional” language. Additional disclaimers may be added as required; for example, nonendorsement of commercial products and services (see all disclaimers at <https://www.usgs.gov/about/organization/science-support/office-science-quality-and-integrity/fundamental-science-5>). Note that the names and casing for files are specific, as follows: LICENSE.md, DISCLAIMER.md, code.json, and README.md.
7. Initiate the domain review. This review can be paired with any associated journal article or USGS series publication review. Create an issue on GitLab and label it as “domain review.” Again, keep it open until the review is complete, and use comments to track recommended changes, review completion, and other information as necessary.
8. Initiate the general code review. If there is no associated manuscript, this can be combined with the domain review (see number 7) if the reviewer has sufficient knowledge. Create an issue on GitLab and label it as “code review.” Keep the issue open until the review is complete, and use comments to track recommended changes, review completion, and so forth.

6 Best Practices for Distributing and Deploying USGS Shiny Applications

9. Initiate the administrative security review(s). These reviews must be completed on all versions of the source files in the repository, including any changes that are made after the initial security review. It may be easier to perform security reviews incrementally with each commit, depending on workflow. To initialize reviews, create an issue on GitLab (see <https://docs.gitlab.com/ee/user/project/issues/>), label the issue as “administrative security review,” and keep it open until review has been completed. Once the review is complete, close the issue. Reopen the same issue for any future security reviews. Use the comments feature to track review completion, recommended changes, and any other pertinent information.
10. Once all reviews are completed and Bureau approval within IPDS has been obtained, create a candidate release branch with a version number. Next, update disclaimers in the release branch using the “approved” language and, finally, create an annotated tag using the final commit on the release candidate branch as the basis. Note that, unlike commits, tags must be explicitly pushed to GitLab using the command `Git push origin <tagname>`, where `<tagname>` is the name of the tag.
11. Attach any necessary files to the tag.
12. Submit a release issue referencing the tag to the USGS CHS software release group on InnerSource GitLab (note: authors of provisional software releases with Science Center Director approval can request public releases from this page as well). Cloud Hosting Solutions will work with users to perform the final steps necessary for public release.

The remote repository will be made publicly accessible by USGS CHS once they review the repository and ensure that the project has received all other required approval (IPDS, all reviews, and other approval). Any files (zipped application, documentation, license, and other required documents) attached to the release tag will also become publicly accessible. Authors should verify that a permanent digital object identifier has been reserved for their software release (often performed by the Center’s data management team). If the application is hosted on the web, it can be made public at this stage.

Data Releases Associated with Shiny Applications

New USGS data products contained within Shiny applications will need to undergo a data release in parallel with the software. The data release process is applicable when the software uses data created as a USGS science product, or funded by USGS, that are not already approved and available from public databases or repositories. Open-file formats are preferred. A full description of the requirements for USGS

data releases can be found on the USGS data management website (<https://www.usgs.gov/data-management/data-release>).

It is helpful to engage the data management team early in the project to ensure best practices are followed for data release requirements. To initiate a data release, authors must understand all requirements and prepare appropriate metadata before the review from the data management team. There are many useful tools available to aid in metadata creation (for example, the USGS Metadata Wizard 2.x; <https://code.usgs.gov/usgs/fort-pymdwizard>). These tools are available at the USGS data management site (<https://www.usgs.gov/products/data-and-tools/data-management/metadata-creation>). Note that special symbols (for example, %, –) are not supported in the Extensible Markup Language (XML) metadata documents. The Metadata Wizard can automatically extract feature information, including coordinate system and spatial extent in spatial data and column headers in tabular data, but authors need to fill in attribute information fields.

Once the data and associated metadata are finalized, authors can engage with the data management team to begin the official review. Data releases are IPDS products that require data and metadata reviews and Science Center Director approval. Data releases will typically be hosted through ScienceBase, but there are other approved repositories (see <https://www.usgs.gov/data-management/repositories>). Appropriate metadata and Bureau approval within IPDS are still required when using third party repositories, and a metadata record will be sent to the Science Data Catalog (SDC) when alternative repositories are used.

Shiny Application Requirements and Best Practices

Standards for Visual Appearance

Like all web-based USGS applications, Shiny applications that will be publicly deployed to the web are required to follow the USGS visual identity standards template for web-based applications (fig. 2), which includes an approved USGS header and footer. This template is made available through the USGS Office of Communication and Publishing (OCAP) Digital Services, and users should check for updates before publication of web applications.

Not all components of the template are required (for example, the search bar and social media links can be hidden). However, the USGS logo and footer links must be present. Additionally, the header and footer should stretch across the full width of the page.

Although there are several ways to apply the visual identity template in Shiny applications, one straightforward approach is to save the header and footer templates as HTML documents in the Shiny application’s “www” folder. Then, they can be included in the application by calling them from the user interface (UI) script. Here is an example:

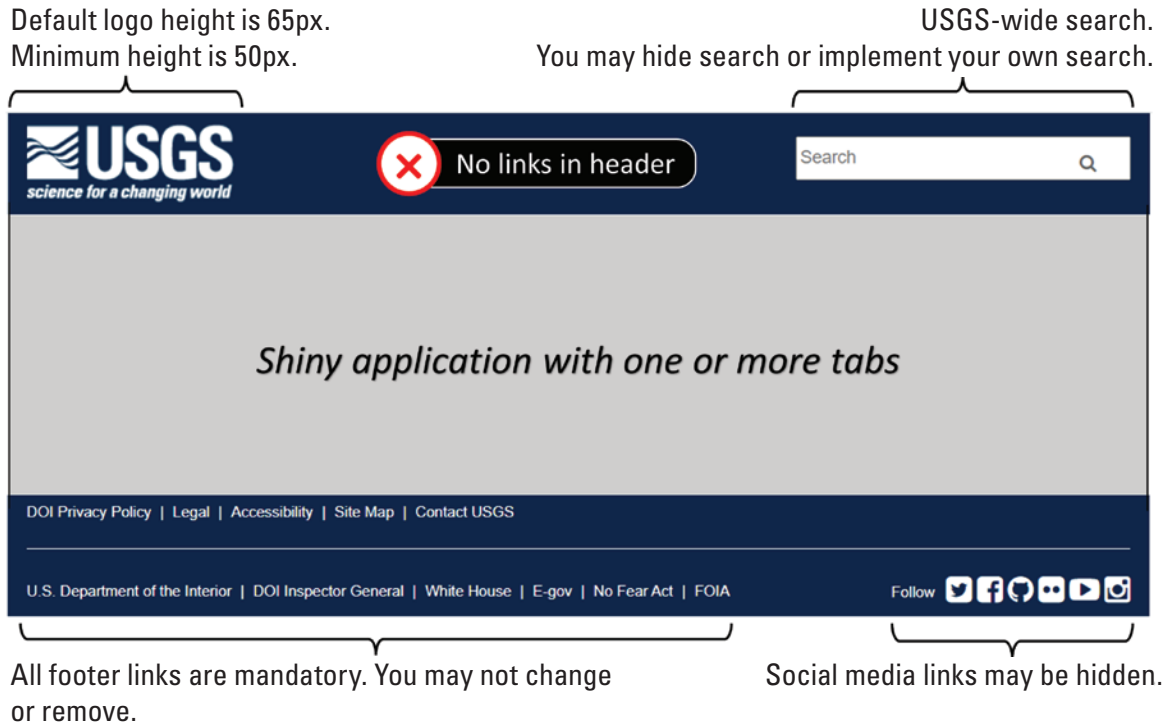


Figure 2. U.S. Geological Survey (USGS) appearance requirements for web pages (image modified from the Office of Communication and Publishing [OCAP] visual identity web application template).

```
# header absolutePanel(
tags$head(tags$style( includeHTML("www/
header.html") ) )
```

If the application is being distributed as an R package (R project for Statistical Computing; <https://www.r-project.org/>), the HTML documents can be saved in the “inst” folder and called using the `system.file()` and `htmlTemplate()` functions.

Disclaimers

All Shiny applications made available to the public must include the appropriate software disclaimer text (see the FSP—Guidance on Disclaimer Statements at <https://www.usgs.gov/about/organization/science-support/office-science-quality-and-integrity/fundamental-science-5>). The required text differs based on whether an application is provisional or approved, and scientific or non-scientific. Approved Scientific Software applications will need to include the disclaimer for Scientific Software:

“This software has been approved for release by the U.S. Geological Survey (USGS). Although the software has been subjected to rigorous review, the USGS reserves the right to update the software as needed pursuant to further analysis and review. No warranty, expressed or implied, is made by the USGS

or the U.S. Government as to the functionality of the software and related material nor shall the fact of release constitute any such warranty. Furthermore, the software is released on condition that neither the USGS nor the U.S. Government shall be held liable for any damages resulting from its authorized or unauthorized use.”

However, while in the development stage (before approval and release), the preliminary language must be used:

“This software is preliminary or provisional and is subject to revision. It is being provided to meet the need for timely best science. The software has not received final approval by the U.S. Geological Survey (USGS). No warranty, expressed or implied, is made by the USGS or the U.S. Government as to the functionality of the software and related material nor shall the fact of release constitute any such warranty. The software is provided on the condition that neither the USGS nor the U.S. Government shall be held liable for any damages resulting from the authorized or unauthorized use of the software.”

The disclaimer text can easily be included on Shiny applications using one of the supported HTML tags. For example, the `p()` tag will add text directly to the page. There are also commands for including R markdown documents (`includeMarkdown()`) and text files (`includeText()`).

Deployment of Shiny Applications to the Web

There is no Bureau-wide standard for deployment of software over the web. However, multiple options exist for deploying and distributing Shiny applications.

For Shiny applications requiring public web access, USGS CHS has made available RStudio Connect/Posit, a cloud-based publishing platform for Shiny applications. Through this service, users may deploy Shiny applications to the web through RStudio desktop. Alternatively, applications may be deployed to RStudio Connect/Posit via USGS OpenSource GitLab, allowing direct deployment of approved Scientific Software applications. Users should note that, although RStudio Connect/Posit is a public-facing environment, applications deployed for public visibility must first be approved by following one of the established procedures outlined in this document. Licenses for RStudio Connect/Posit will be provisioned through USGS CHS and are available upon request.

When a public-facing website is not required, Shiny applications can be released as an R package on CRAN if the authoritative source code has first been released on USGS OpenSource GitLab. Shiny applications can also be configured to run from online sources. For example, Shiny applications can be deployed locally from a uniform resource locator (URL) if the target audience is able to use RStudio and install R packages on their computer. The Shiny package provides a command to download and run Shiny applications (distributed as a .zip file) from a downloadable web link: `runURL()`. This command will download and launch a Shiny application (stored as a .zip file) within RStudio on a user's personal computer. Many R functions can also download data from a URL, which enables Shiny applications to draw from external data sources.

Testing

Make sure to thoroughly test the application to avoid last minute changes and hold ups. Remember that all changes performed after the administrative security review will require an additional security review. When possible, write unit tests for individual functions (see the section on testing in the USGS documentation at <https://www.usgs.gov/products/software/software-management/testing-and-automation>). Shiny applications may involve several dependencies between reactive objects in the R backend and user inputs on the user interface. For this reason, it is important to test that all HTML components (including action buttons, file menus, numeric input fields, and user uploads) respond and update correctly after user interaction. Shiny applications will frequently crash if reactive components and their dependencies are not properly defined in the server script.

Error Messages

Default Shiny error messages can be confusing and uninformative. Use the `validate()` and `showNotification()` functions to create helpful messages that allow the application to fail gracefully and warn the user in an informative way. Applications that include user input data, such as coordinates, can include validation steps to inform the user if input files are improperly formatted or outside the area of coverage.

Documentation

Documentation of source code is a required component of software development and must be included when developing Shiny applications as Scientific Software. Although documentation of source code is required to be present in the authoritative GitLab repository (see “[Software Release Process for Shiny Applications](#)”), authors of Shiny applications that will be deployed on a public website need to make online documentation available so that users can understand the science behind an application on the website. If an application is associated with another USGS product (for example, a journal article or data release), this online documentation is a good place to link to these associated resources.

One of the easiest ways to provide documentation is to include a help page as a separate tab in a Shiny application. For example, an HTML help file can be written in R Markdown and stored as an HTML page. The `includeHTML()` command can then place the help document on a `tabPanel()` in the application.

Versioned Updates to Shiny Applications

If the Shiny application requires updates after the approved release, changes can be made in a provisional software release if there are no substantial scientific changes. Such changes could include the addition of new data or new features to the application. Whether or not planned changes require the full review process is to be determined by the PI and their supervisor, with Science Center Director approval. As always, any changes to source code in the provisional release is subject to administrative security reviews.

The frequently asked questions on the USGS software management page provides additional detail on the different types of versioned updates. These include patches, minor revisions, and major revisions. Major revisions require a new IPDS approval record from the Science Center Director. The following is a suggested workflow for updating software:

1. Create a new branch in the GitLab project repository to hold any changes from the previously approved version.

2. Make the necessary software updates in the new working branch, including any changes to the software documentation and metadata. Update the software disclaimers in this branch to use the language for provisional Scientific Software (see “[Disclaimers](#)”).
3. In consultation with the PI, Research Manager, and Science Center Director, determine the level of review necessary and assign reviewers using issue tracking. The level of review will depend on whether the update is a patch, minor version, or major version. IPDS tracking and approval is required for major version updates.
4. When reviews are complete and the new version has been approved in IPDS (if it is a major version), update software disclaimers. Next, create a new release tag for the updated branch in your GitLab project, assigning an appropriate version number to the tag and any other associated descriptions or files.

Summary

- Shiny applications intended for public release are required to go through the U.S. Geological Survey (USGS) software release process.
- Shiny applications that implement scientific algorithms or produce scientific data must be reviewed and approved as Scientific Software.
- If the application produces new interpretive information, it must be paired with a USGS publication series or journal article.
- Approved Shiny software applications can be deployed to the web for public use if they include all required USGS visual standards and disclaimers.

Acknowledgments

This document has been vastly improved by comments and suggestions from several helpful reviewers. In particular, we thank Julie Yee, Robert Klinger, Cali Roth, Melanie Davis, Isa Woo, Eric Martinez, and Michelle Jeffries of the U.S. Geological Survey for their insight.

Glossary

Git A version control software for source code.

GitLab A web-based Git-repository manager.

Scientific software Software containing source code implementing scientific algorithms or producing scientific data.

Shiny A web-based application coded in R using the 'Shiny' package.

Software Computer programs that comprise a series of instructions, rules, routines, or statements, regardless of the media in which recorded, that allow or cause a computer to perform a specific operation or series of operations.

For more information concerning the research in this report,
contact the

Director, Western Ecological Research Center

U.S. Geological Survey

3020 State University Drive East

Sacramento, California 95819

<https://www.usgs.gov/centers/werc>

Publishing support provided by the U.S. Geological Survey

Science Publishing Network, Sacramento Publishing Service Center

