

# Software Deposit: How to deposit software

Michael Jackson (ed.), The Software Sustainability Institute

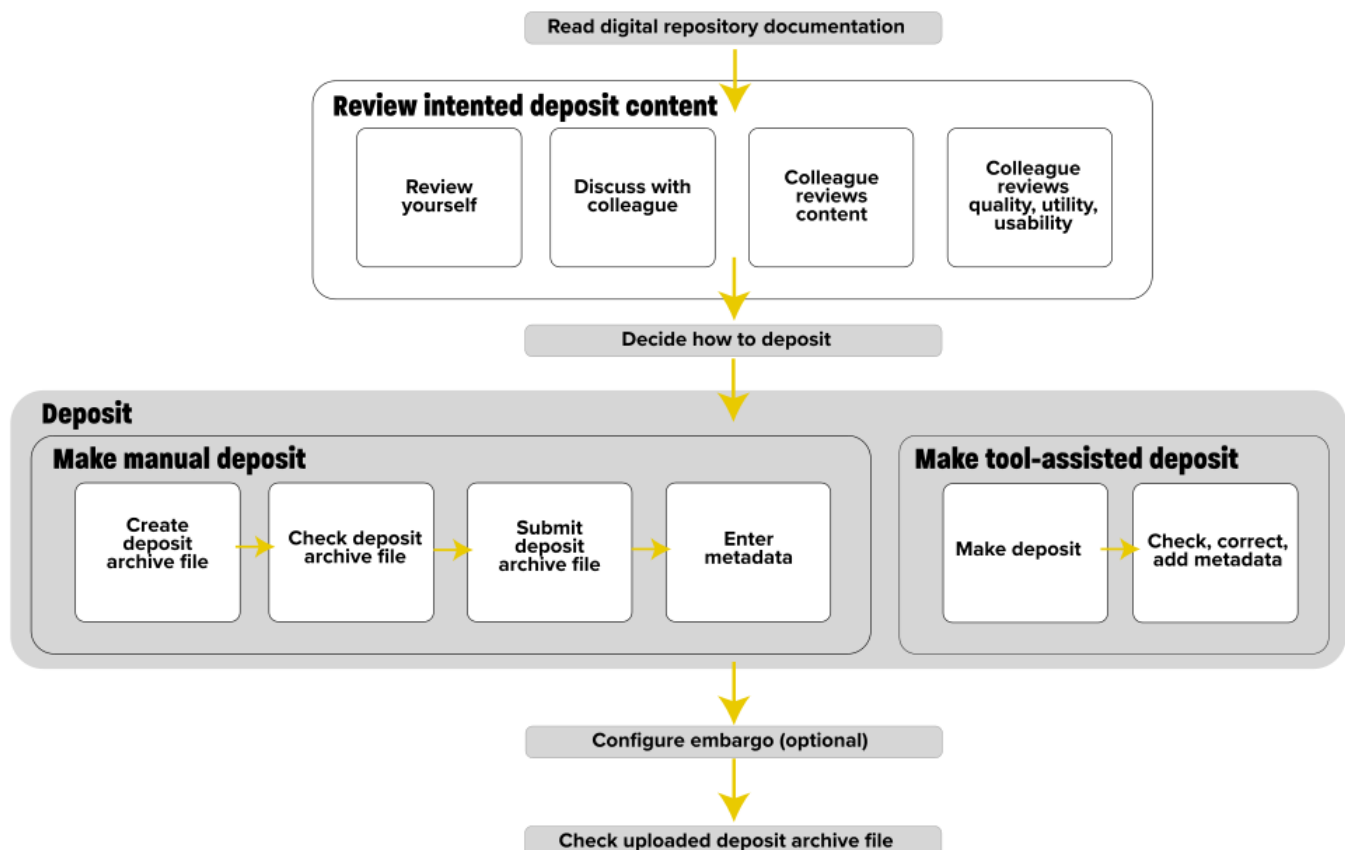
Version 1.0

doi:[10.5281/zenodo.1327327](https://doi.org/10.5281/zenodo.1327327)

07 August 2018

## Introduction

You know why you want to deposit your software, have chosen when to do it and where and know what you need to deposit and how to describe it. Now you are ready to deposit your software in your chosen digital repository. Different digital repositories have different means of submission and different requirements as to the deposits they will accept, the metadata associated with these deposits and how deposits are done. However, regardless of the digital repository you are using, there are some common tasks that you should do before, during and after you deposit your software, which are described in this guide.



## How to deposit software

## About this guide

This guide is one of a series of guides on software deposit, written by The Software Sustainability Institute [1](#),

funded by Jisc <sup>2</sup>. For an overview of the series, see Michael Jackson (ed.) (07 August 2018). Software Deposit: Guidance for Researchers (Version 1.0). Zenodo. doi:[10.5281/zenodo.1327310](https://doi.org/10.5281/zenodo.1327310). Online: <https://softwaresaved.github.io/software-deposit-guidance/SoftwareDepositGuidance.html>.

## **Read the documentation provided by the digital repository**

Your chosen digital repository should provide documentation that describes exactly how you can make your deposit, including what you can deposit, the supported file formats for deposits, what metadata you may need to provide, how to provide this metadata and what happens after you make your deposit. If you're lucky, they'll have an intuitive user interface that informs you of this as you make your deposit.

If there is any conflict between the advice and guidance in our guides and in your chosen digital repository, then the requirements of the digital repository take precedence.

## **Review the intended deposit content**

There are a number of ways you can review your intended deposit content.

### **Review the intended content yourself**

You should always review your intended content yourself, checking that it will include everything you intend to deposit.

If applicable, check that your software can be built, deployed, installed and run following your user documentation. Take care that your environment and your familiarity with the software do not cause you to skip steps or make assumptions that someone having a different environment or not familiar with the software would not make.

### **Discuss the intended content with a colleague**

Discuss with a colleague what your software does and how it does it, give them a demonstration of it in action and walk them through the resources you plan to include in your deposit. Discussing your software and deposit with a colleague may reveal assumptions you have made that are wrong or you have assumed and that need to be made explicit.

### **Ask a colleague to review the intended content**

Ask a colleague to check that your deposit will include everything you intend to deposit, remembering to let them know what you intend to deposit.

### **Ask a colleague to review the quality, utility and usability of the intended content**

Ask a colleague to try to build, deploy and run the software, following its user and developer documentation, reviewing these for readability, accuracy and usability. While the downside of such a review is that it requires a lot more time to do, it provides the most comprehensive feedback on your software <sup>3</sup>.

## **Decide how to deposit**

Most digital repositories support manual deposits by means of completing web-based forms and selecting the files you want to deposit.

However, your digital repository may support tools which provide some form of automation for this process, depending upon where your source code, and related resources, are held. For example, if using GitHub and intending to deposit into Zenodo or figshare, then the figshare-GitHub integration <sup>4</sup> or Zenodo-GitHub

integration [5](#) can automate much of the deposit process for you.

## Make a manual deposit

### Create a deposit archive file

Create a directory containing the contents of your deposit.

Give the directory a meaningful name, e.g., "your-software-name/". If you have a software release process and use semantic versioning [6](#) then name the directory using your version numbering scheme, e.g., "your-software-name.1.0.0/".

Archive your directory into an archive file. Good formats to use for your archive file are ".zip", ".tar", ".tar.gz", ".tgz", or ".tar.bz2", all of which are popular open formats. Avoid using ".7z" or ".rar", as support for these is not available by default in current versions of Linux, Windows or MacOS.

Give your archive file a name that matches your directory name and file extension that describes the archive format e.g. "your-software-name.zip", "your-software-name.tar", "your-software-name.tar.gz", "your-software-name.tgz", "your-software-name.tar.bz2", or, if using semantic versioning, "your-software-name-1.0.0.zip".

### Check the deposit archive file

Copy your deposit archive file to a separate part of your file system. Unpack the deposit archive file to ensure that it has not been corrupted and can be unpacked successfully.

Check that the directory content matches the original directory that you archived.

Now is a good time to do a final check to ensure that the deposit contains everything you want to deposit and does not contain anything that you don't want to deposit. If anything is missing or anything is there that shouldn't be, then make the changes to your original deposit directory and try again.

### Submit the deposit archive file

Following the documentation of your chosen digital repository, submit your deposit archive file to the digital repository.

If the digital repository allows for multiple files to form a single deposit, then, in addition to your deposit archive file, deposit your README and, if you have one, your codeMeta.json file, so others can easily access and view these, without having to unpack your deposit archive file.

### Enter metadata

Enter metadata requested by the digital repository. Ensure that the metadata you enter is consistent with any metadata that you have specified within your deposit (e.g. in your LICENCE or codeMeta.json files).

Beware default values assumed by the digital repository and check that these are what you want. If not, then update their values in the digital repository. You don't want to end up with the digital repository metadata recording a difference software licence from the one you actually want, for example.

## Make a tool-assisted deposit

Follow the instructions for your chosen deposit tool to make an automated deposit. For example, when using the figshare-GitHub integration or Zenodo-GitHub integration, a deposit is triggered by creating a new release of your software within GitHub.

## Check, correct and add metadata

Tools to help automate software deposits may be limited in the metadata they can extract and provide to the digital repository. You may need to add additional metadata directly, using a web-based interface provided by the digital repository, for example.

These tools may also infer incorrect values for metadata. Ensure that the metadata that has been inferred is both correct and consistent with any metadata that you have specified within your deposit (e.g. in your LICENCE or codeMeta.json files, your contributors or the primary programming language used to implement your software) and make any necessary corrections to the values in the digital repository

Look out for default values assumed by the digital repository and check that these are what you want. If not then update their values in the digital repository. You don't want to end up with the digital repository metadata recording a different software licence from the one you actually want, for example.

You may also want to add additional metadata beyond that which was automatically inferred.

## Enforce an embargo (optional)

If you want to enforce an embargo and the digital repository supports this feature, set the date until which the deposit is to be embargoed and provide any additional information required by the digital repository.

## Check your uploaded deposit

Once uploaded and, if applicable, published, download and unpack your deposit archive file, following the process described above.

## Find out more

If you are depositing into an institutional repository, your local research data manager will be able to advise you on how to make your deposit. Your colleagues may already have experience in making deposits and also be able to offer help and advice.

Related Software deposit guides:

- Michael Jackson (ed.) (07 August 2018). Software Deposit: What to deposit (Version 1.0). Zenodo. doi:[10.5281/zenodo.1327325](https://doi.org/10.5281/zenodo.1327325). Online: <https://softwaresaved.github.io/software-deposit-guidance/WhatToDeposit.html>.
- Michael Jackson (ed.) (07 August 2018). Software Deposit: What not to deposit (Version 1.0). Zenodo. doi:[10.5281/zenodo.1327323](https://doi.org/10.5281/zenodo.1327323). Online: <https://softwaresaved.github.io/software-deposit-guidance/WhatNotToDeposit.html>.
- Michael Jackson (ed.) (07 August 2018). Software Deposit: How to describe a software deposit (Version 1.0). Zenodo. doi:[10.5281/zenodo.1327321](https://doi.org/10.5281/zenodo.1327321). Online: <https://softwaresaved.github.io/software-deposit-guidance/HowToDescribeSoftwareDeposit.html>.
- Michael Jackson (ed.) (07 August 2018). Software Deposit: How to review a software deposit (Version 1.0). Zenodo. doi:[10.5281/zenodo.1327314](https://doi.org/10.5281/zenodo.1327314). Online: <https://softwaresaved.github.io/software-deposit-guidance/HowToReviewSoftwareDeposit.html>.

Archive naming:

- Raymond, E.S. (2000) Software Release Practice HOWTO, <http://www.tldp.org/HOWTO/Software-Release-Practice-HOWTO/>, has a good section on naming software archive files, <http://www.tldp.org/HOWTO/Software-Release-Practice-HOWTO/naming.html>.

## Cite this guide

Please cite as: Michael Jackson (ed.) (07 August 2018). Software Deposit: How to deposit software (Version 1.0). Zenodo. doi:10.5281/zenodo.1327327. Online: <https://softwaresaved.github.io/software-deposit-guidance/HowToDepositSoftware.html>.



This work is published under a Creative Commons Attribution 4.0 International License (CC BY 4.0), <https://creativecommons.org/licenses/by/4.0/>.

---

1. The Software Sustainability Institute, <https://www.software.ac.uk>.↵
2. Jisc, <https://www.jisc.ac.uk>.↵
3. This is a good thing to do regularly, regardless of whether you are depositing software or not. It can help to increase your software's bus factor ([https://en.wikipedia.org/wiki/Bus\\_factor](https://en.wikipedia.org/wiki/Bus_factor)).↵
4. "How to connect figshare with your GitHub account", figshare knowledge, <https://knowledge.figshare.com/articles/item/how-to-connect-figshare-with-your-github-account-1>.↵
5. "Making your code citable", GitHub Guides, <https://guides.github.com/activities/citable-code/>.↵
6. Semantic Versioning, <https://semver.org/>.↵