# Software Deposit: What not to deposit

Michael Jackson (ed.), The Software Sustainability Institute

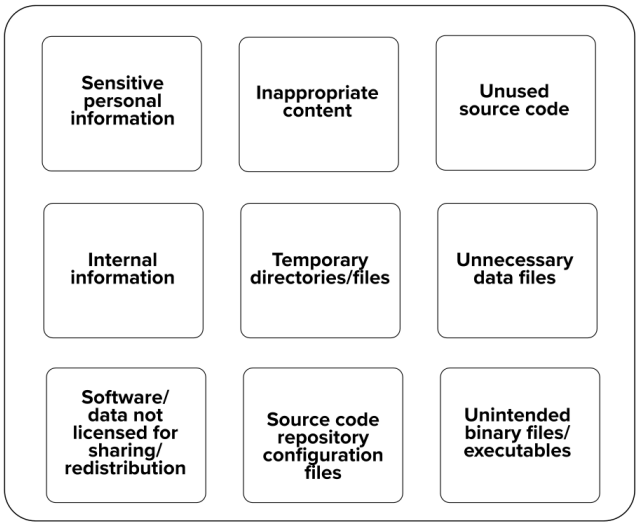
Version 1.0

doi:10.5281/zenodo.1327323

07 August 2018

## Introduction

A software deposit lodged within a digital repository can contain myriad content. But there is certain content that should not be included within a software deposit. Some of this content may be innocuous and only result in your deposit being more bloated than it otherwise needs to be. Some of this content may compromise your security. And, in the worst case, some of this content may result in you inadvertently breaking local laws relating to data protection. This guide summarises the content that should not be deposited with your software.




**What not to deposit**

## 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 [2]. 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. Online: https://softwaresaved.github.io/software-deposit-guidance/SoftwareDepositGuidance.html.

## Sensitive personal information

Check that your content does not contain any personal or identifiable information about other people which, if shared, may result in you violating local laws relating to data protection or patient confidentiality. This

information can include: names, addresses, email addresses, phone or fax numbers, photos, gender, sexuality, sexual preferences, political beliefs, religious beliefs, allergies, illnesses, disabilities, financial data and medical data.

For names, email addresses and organisational information relating to those who have written or contributed to your software, make sure you have received their permission to include this information.

Look for sensitive personal information in documentation, source files, data files and image files (especially medical image files).

# Internal information

Check that your content does not include any information relating to your local infrastructure and servers and which might compromise their security if this information were exposed publicly. This information can include: URLs or IP addresses of servers or service endpoints, usernames, passwords, private SSH keys, security tokens and credentials.

Look for internal information in documentation, configuration files, data files and source code.

# Software and data not licensed for sharing or redistribution

If you have built your software using third-party components then it may include source code, libraries or binaries which you are not allowed to redistribute. This is often the case with proprietary software from commercial companies. Similarly, you may not be allowed to redistribute certain data files. This is often the case with data that has been shared with you but is not for publication or redistribution to a third-party. Redistributing such software or data may cause you to violate confidentiality agreements with collaborators or local laws relating to copyright and intellectual property.

Check that the licences of any third-party software or data files allow you to redistribute them. If not, then remove these files and, instead, document them as dependencies.

# Inappropriate content

Remember that witty and profane comment you buried in your source code after battling a day to fix a bug? Or that amusingly tasteless cartoon image you downloaded, but can't remember where? Take care that your deposit does not include such content as it will be associated with your name in a digital repository for quite some time and may cause, at best, amusement to your colleagues and, at worst, damage to your reputation.

Look for inappropriate content in documentation, data files, source code (especially comments), images and movies.

# Temporary directories and files

Many software packages, including your own, may create temporary directories and files. While useful in your day-to-day work, these won't be needed by others wishing to use your software and their presence in your deposit will unnecessarily bloat it and make its content look cluttered.

Check that your content does not include temporary directories or files that others do not need to understand or run your software. These can include: files starting or ending in "~"; XEmacs scratch files (e.g. "Dft.java~"); Microsoft Word, Excel and Powerpoint temporary files (e.g. "~wrcxxxx.tmp"); and files or directories called, or ending in "tmp".

# Source code repository configuration files

If you are using a source code repository and are creating a software deposit from it, then make sure that you

do not include the repository configuration files and directories within your deposit. Their presence in your deposit will unnecessarily bloat it.

These files and directories can include ".git/" directories, ".hg/" directories, ".svn/" directories and "CVS/" directories. Git, Mercurial, Subversion and CVS all provide commands to get a copy of the source code repository without these repository configuration files and directories [3].

Tools that help deposit software from source code repositories into digital repositories, such as the figshare-GitHub integration [4] and Zenodo-GitHub integration [5], ignore these repository configuration files automatically.

# Unused source code

Source code that is no longer used can unnecessarily bloat your deposit. It can also make it challenging for others to understand what your software does as they will assume, naturally, that all the source code in your deposit serves some purpose.

Remove any source code files that are no longer used. Similarly remove any commented-out code.

# Unnecessary data files

Including sample data files can be valuable for those who use your software deposit. These provide users with sample inputs, to use with your software, and the expected outputs, against which they can compare their outputs when running your software. However, data files can also increase the size of your deposit.

Check your data files and ensure you are only depositing data files that are of use to others, for example sample inputs and the corresponding outputs.

If you have published, or intend to publish, results based on your data files, then consider following the FAIR [6] principles for research data management and deposit them into a digital repository as a citable research object too. You can then use the metadata that describes your software and data deposits to link them together [7].

# Unintended binary files and executables

Check that the only binary files and executables you include in your deposit are those you are explicitly and intentionally choosing to include. Any that you don't consider to be needed by others wishing to use your software will only unnecessarily bloat it.

# Find out more

Related Software deposit guides:

- Michael Jackson (ed.) (07 August 2018). Software Deposit: What to deposit (Version 1.0). Zenodo. doi:10.5281/zenodo.1327325. Online: https://softwaresaved.github.io/software-deposit-guidance/WhatToDeposit.html.
- Michael Jackson (ed.) (07 August 2018). Software Deposit: How to choose a software licence (Version 1.0). Zenodo. doi:10.5281/zenodo.1327316. Online: https://softwaresaved.github.io/software-deposit-guidance/HowToChooseSoftwareLicence.html.

Temporary directories and files:

- "A collection of .gitignore templates", GitHub, https://github.com/github/gitignore. A list of templates for various programming languages and software development framework. Each of these lists the types of files that are typically not to be added to source code repositories. These can serve as a guide

for the types of files not to be included in a software deposit too.

Depositing research data:

- DCC (2014). "Five steps to decide what data to keep: a checklist for appraising research data v.1". Edinburgh: Digital Curation Centre. Available online: http://www.dcc.ac.uk/resources/how-guides/five-steps-decide-what-data-keep
- Whyte, A. (2015). "Where to keep research data: DCC checklist for evaluating data repositories" v.1.1 Edinburgh: Digital Curation Centre. Available online: http://www.dcc.ac.uk/resources/how-guides-checklists/where-keep-research-data/where-keep-research-data

# Cite this guide

Please cite as: Michael Jackson (ed.) (07 August 2018). Software Deposit: What not to deposit (Version 1.0). Zenodo. doi:10.5281/zenodo.1327323. Online: https://softwaresaved.github.io/software-deposit-guidance/WhatNotToDeposit.html.



---

1. The Software Sustainability Institute, https://www.software.ac.uk.↩
2. Jisc, https://www.jisc.ac.uk.↩
3. For example, "git archive", https://git-scm.com/docs/git-archive; "hg archive", https://www.mercurial-scm.org/repo/hg/help/archive; "svn export", http://svnbook.red-bean.com/en/1.7/svn.ref.svn.c.export.html; "cvs export", https://www.gnu.org/software/trans-coord/manual/cvs/html_node/export.html.↩
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. "The FAIR Data Principles", FORCE 11, https://www.force11.org/group/fairgroup/fairprinciples.↩
7. For example, CodeMeta, https://codemeta.github.io/terms/, provides a "supportingData" term; figshare, https://figshare.com/, provides a "references" field; and, Zenodo, https://zenodo.org, provides a "relatedIdentifiers" property.↩