

## Handling Version Information

[support@rsp.ac.uk](mailto:support@rsp.ac.uk)

### Overview

Academic researchers typically produce many revisions of a research output, up to 60 in some cases. By adding version information to papers, with clear links to published versions, it is possible to allay concerns about the quality of different versions. The briefing paper provides recommendations for handling versions within digital repositories.

**The VERSIONS Project** conducted a study, including two surveys and 26 interviews, to address the uncertainties relating to versions of academic papers in repositories.

**The Version Identification Framework (VIF)** provides guidance and solutions for repository managers, content creators and software developers about identifying versions of any type of digital object.

### **VERSIONS: User requirements and attitudes about versions**

The VERSIONS user requirements study highlighted some clear trends. Academic researchers typically produce many revisions and versions of a research output, up to 60 in some cases, and it increases when working with co-authors. Authors vary in how many of these versions they keep: either all, milestone only, or just the latest. They tend to want to limit the number of versions disseminated. To help reduce uncertainty, VERSIONS recommends the terms shown in Table 1 for describing journal articles. A key message for repository managers to convey is that, because of standard agreements between authors and publishers, keeping only the Published Version is likely to limit the author's ability to make their work accessible in future.

As a minimum keep milestone versions like an author's Accepted Version.

Half of authors are not satisfied with the way they manage their personal collections of digital objects.

An institutional repository provides academic staff with a managed environment to deposit their milestone versions, where they can be easily located and retrieved.

<b>Draft</b>	Early version circulated as work in progress
<b>Submitted Version</b>	The version that has been submitted to a journal for peer review
<b>Accepted Version</b>	Author-created version incorporating referee comments and is accepted for publication
<b>Published Version</b>	The publisher-created published version
<b>Updated Version</b>	A version updated since publication

Table 1. Recommended terms for describing versions of journal articles

81% of authors surveyed stated they would deposit their final Accepted Versions of journal articles in an institutional repository 'if invited to do so'. Adding version information to authors' papers, with clear links to published versions, helps allay concerns about the quality of accepted versions and about potential loss of downloads for published versions.

Institutional repository managers can help inform academic authors about publishers' access policies. Researchers remain uncertain and unaware of possible negotiable alternatives. SHERPA RoMEO is an excellent resource for providing detailed advice to authors on 'permitted' versions.

## Advice on versions for authors

The VERSIONS toolkit offers 5 top hints for authors, derived from feedback by active researchers:

1. Plan how you will store and name your personal versions of files.
2. Keep permanently your own author-created Submitted Versions and final author-created Accepted Versions of research publications.
3. Add the completion date to the first page of any versions, especially milestone versions.
4. Consider carefully how to disseminate your work before signing any agreements with publishers. Keep a copy of your signed agreements.
5. Deposit your work in an open access repository and guide readers to your latest and published versions.

## VIF: The importance of versioning

As repositories have grown in size and scope, the problems caused by versioning have become more prominent. It is now an accepted problem; only 5% of academics and 6.5% of information professionals find it easy to identify versions of digital objects within institutional repositories. Across multiple repositories this becomes 1.8% and 1.1% respectively.

Version identification is not about identifying the 'best' or 'right' version of an object. VIF aims to make the version status of an object and the relationships between linked objects clear to end users so they can identify which is the most appropriate version for their purpose.

## Version information and version relationships

Versioning problems are minimised if some or all of 5 key pieces of information that exist at the time of creation of the object are captured, defined and made sufficiently visible to the user, to a deposit mediator or made machine-readable:

- Defined dates
- Identifiers

- Version numbering
- Version labels / taxonomies
- Text description

It is important to include versioning information both in metadata and within an object itself. There are ways of accessing an object, e.g. directly through a search engine like Google, which can bypass its associated metadata, but if the information is also embedded within the object, it cannot be lost and the user can determine what version they have at hand. VIF recommends at least one of the following solutions is used systematically within a repository:

- The filename
- A watermark
- A coversheet
- An ID tag or property field

Clarifying the relationship between versions is equally as important and using a FRBR (Functional Requirements for Bibliography Records) entity-relationship model to structure metadata helps to do this. The structure shows where a copy of an object fits into the full span of different iterations, formats, and expressions of a work.

## VIF guidance

VIF is a detailed information resource with sections to highlight the problem of versioning and the latest research available. It explores essential versioning information and gives more detail about the recommendations discussed above.

Repository managers might find the strategy checklist a useful place to start with VIF. Some thought and discussion about the issues and how these affect the repository will save much effort later.

VIF also has dedicated sections for content creators, giving tips, guidance and recommendations for software development.

## References & Further information

SHERPA/RoMEO

<http://www.sherpa.ac.uk/romeo/>

SPARC Resources for Authors

<http://www.arl.org/sparc/author/index.shtml>

Version Identification Framework

<http://www.lse.ac.uk/library/vif/>

VERSIONS Project <http://www.lse.ac.uk/versions>

Repository Support Project <http://www.rsp.ac.uk/>