

# Digital preservation and curation

## The danger of overlooking software

From preserving research results, to storing photos for the benefit of future generations, the importance of preserving data is gaining widespread acceptance. But what about software?

It's easy to focus on the preservation of data and other digital objects, like images and music samples, because they are generally seen as *end products*. The software that is needed to access the preserved data is frequently overlooked in the preservation process. But without the right software, it could be impossible to access the preserved data - which undermines the reason for storing the data in the first place.

This briefing paper is targeted at people who are responsible for preserving data and digital objects on the behalf of others. These people typically work for libraries, museums and archives.

Our goal in this paper is to explain why long-term software preservation is necessary, what needs to be understood before software can be preserved and how to get started with the preservation process.

### When should you consider software preservation?

Software is used to create, interpret, present, manipulate and manage data. You should consider software preservation whenever one or more of the following statements is true:

#### 1. The software can't be separated from the data or digital object

In an ideal world, data can be isolated and preserved independently of the software used to create or access it. Sometimes this is not possible. For example, if the software and the data form an integrated model, the data by itself is meaningless. This means that the software must be preserved with the data.

If data is stored in a format that is open and human-readable, then any software that follows that format can be used to read the data. If the data is stored in a format that is closed and arcane, then you must also preserve the software that is used to access it.

#### 2. The software is classified as a research output

The software could fall under a Research Council's preservation policy. This means that the software must be preserved as a condition of its funding.

#### 3. The software has intrinsic value

Software can be a valuable historical resource. If the software was the first example of its type, or it was a fundamental part of a historically significant event, then the software has inherent heritage value and should be preserved.

### What are the issues?

Software presents some challenges to those who curate, preserve and archive. In particular, software preservation is difficult, because software is sensitive to changes in its environment.

If there is a change to the computer or operating system on which the software runs, the software will often stop working properly. What's more, this change might not cause a catastrophic failure. Although serious, this kind of failure is at least easy to spot. A change to the computer or operating system might only cause a subtle, yet important, change in results. Expert knowledge is needed to fully understand how a software component works and the effect that a change may have.

There is a lot of variation in software: it comes in many different forms, it is written in a bewildering range of



## Key questions to ask yourself

When considering software preservation, you should consider the following questions:

1. Is there still knowledge and expertise to handle and run the software?
2. How authentic does the preserved software need to be?
3. How **adequate** does the preserved software need to be: should it perform exactly as the original, the same but with only minor deviations, or perform the core functionality only?

## Further information and useful resources

**The Software Sustainability Institute: resources, consultancy and collaborative projects for software preservation and sustainability**

<http://www.software.ac.uk>

**Digital Curation Centre: general digital curation resources and help**

<http://www.dcc.ac.uk>

**Blue Ribbon Task Force on Sustainable Digital Preservation and Access: final report**

<http://brtf.sdsc.edu>

**The Significant Properties of Software project - a JISC funded study into what properties are needed to allow software to be systematically preserved**

<http://bit.ly/eF7yNv>

languages and it can be licensed in many different ways. Further difficulties can arise from the increasing use of web services and the cloud. This is where your software is hosted by external organisations - a practise that is becoming increasingly popular. It generally takes a team of experts to understand all the different facets of software and choose the best route for its preservation.

## How should I approach software preservation?

Software preservation should be part of a broader preservation strategy. This strategy should provide a guide of what needs to be preserved, and for how long.

The same considerations that apply to digital preservation also apply to software (intellectual property, choice of media, backup and recovery, etc.), so the basic considerations of software preservation are similar to those of digital preservation. The approaches to preservation are also similar: technical preservation (techno-centric), emulation (data-centric) and migration (functionality-centric).

Preserving the knowledge behind software is as critical as the software itself. Good documentation is important, as is having access to the developers of the software.

A project undertaken by the STFC identified a set of significant properties of software, which can be used as a structured framework to elicit key information from the development team. Developers are keen to contribute to this framework as it helps them organise their documentation and enables their software to live for longer.