

Software preservation

A workshop for digital curators

Matt Shreeve
7 February 2011





Housekeeping

- Fire alarms
- Wireless access to internet
- Mobile phones
- Toilets
- Tea and coffee
- Photography



Two JISC-funded projects



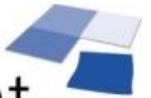
- Significant Properties of Software
 - Identify and analyse significant properties
 - Framework and tool demonstrator
 - 2007 - 2009



Science & Technology
Facilities Council

- Clarifying the purpose and benefits
 - Awareness raising, especially with developers
 - Frameworks and briefing papers
 - 2010

curtis+cartwright



www.software.ac.uk



Four purposes of software preservation

- Encourage software reuse



- Achieve legal compliance and accountability



- Create heritage value



- Enable continued access to data and services





www.software.ac.uk

www.curtiscartwright.co.uk

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



Why are we here today?

- Software sometimes needs to be preserved
 - The software can't be separated from the data or digital object
 - The software is classified as a research output
 - The software has intrinsic value
- Preserving software is hard
 - Very sensitive to changes
 - Many shapes and sizes
- Active engagement with technical developers is necessary