

Objects to collect to preserve software

This is an inclusive list of objects to collect in order to preserve the history of a particular piece of software. (“Software” or “program” as use dhere generically includes a program, an application, a system, etc.)

For no piece of software will all of these objects be collected. For some important software we will try to collect as many as possible. For most software only a few of the objects will be collected.

Each object must be collected with information about its provenance so that historians and researchers can gauge its accuracy and authenticity. We must be aware that in some cases there could be motivation for the contribution of falsified or counterfeit objects.

- The program
 - Copies of the source code intended to be read by people
 - Machine-readable when possible, otherwise on paper which will be scanned
 - Compilable copies of source code.
 - Presumes that the original software development environment is available or can be simulated
 - Depending on the complexity, that also may require collecting
 - Macro libraries, include files, precompiled libraries, etc.
 - Software development tools and files (“make”, “lib”, “link”, etc.)
 - Subroutines, granularity of codes
 - Ready-to-run object code
 - Presumes that the original execution environment is available or can be simulated
 - Packaged or distributed versions of the program
- Collateral materials
 - Documentation
 - User manuals
 - Installation manuals
 - Logic manuals and flowcharts
 - Development notes that elucidate the design process
 - Books and research papers
 - Email Exchanges
 - Contract Materials
 - Marketing material

- Trade show and sales collateral
 - Advertisements
 - Press releases
- Exhibitable objects
 - Card decks, listings
 - Buttons, T-shirts, coffee cups, etc.
- Experiences
 - The developer's experience
 - Interviews (audio, video, transcripts)
 - Photos and videos of people and the environment
 - Personal papers and email
 - Written reminiscences, either contemporaneous or retrospective
 - Identities of all participants
 - Biographies of key people
 - Timeline history of milestones and releases; the product's lifecycle and life history
 - Critical assessments of the software
 - The user's experience
 - Videotapes and photos of the software in operation
 - Reviews and analyses by users and competitors
 - Interviews with early or important users
 - Profiles of typical users
 - The business experience
 - Published reviews
 - Information about the companies involved in development, marketing, sales and support
 - Sales history
 - Competitive environment
- Open Issues:
 - How do we identify the fundamental intellectual ideas in a particular program?
What do we collect to document the evolution of those ideas?
 - How do we preserve the history of universal concepts like "stack" or "B-tree" or LR(k) parsing?