

What's The Trick?

How did we manage to create a programming tool with so much power but at a fraction of the cost of other "interface builders"? We did it by following 5 principles often ignored by other companies:

1. We did not try to redo what Apple has already done well. If you want to edit MENUs, for example, we encourage you to use the excellent MENU editor in Apple's ResEdit program. This contrasts with tools that try to do all things, but do few of these well.

2. We concentrated on improving the weakest parts of the toolbox by providing high-level routines that minimize the size of programs. This contrasts with tools that are burdened with the need to generate voluminous and repetitious code that reflects the existing weaknesses of the toolbox.

3. We made maximum use of existing functionality. ViewIt, for example, extends the Macintosh Control Manager and is therefore compatible with existing controls. This contrasts with tools that make less use of existing toolbox Managers, leading to poorer performance and incompatibilities.

4. We designed our tools to minimize language and compiler requirements. Almost any Mac language, compiler, and programming environment can be used with ViewIt (C, C++, Pascal, Object Pascal, Fortran, Basic, HyperCard, etc.). This provides the widest possible market for our tools, and contrasts with those tied to a single programming paradigm.

5. We minimized overhead. The complete shareware version of ViewIt with on-line documentation and example programs fits on a single 800K disk. This contrasts with tools that are accompanied by many disks and thick manuals that are costly to ship and update.