

Programming Tools Resources Appendix B

B

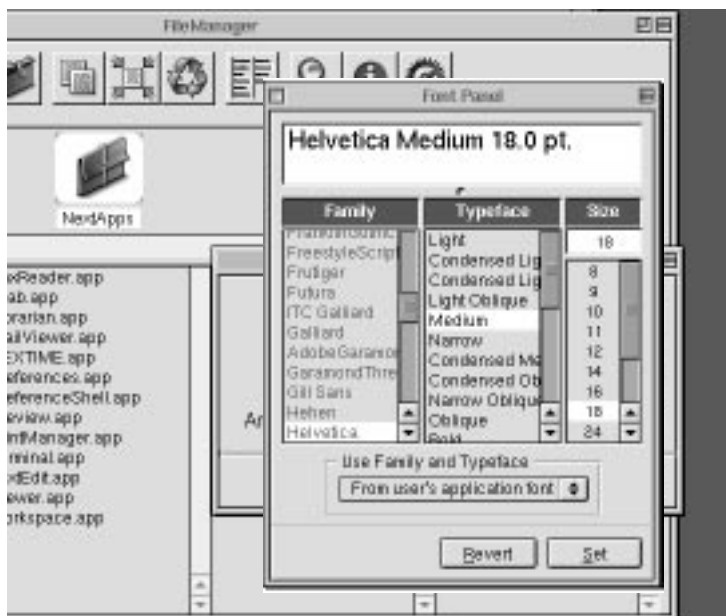
What You'll Learn

Secondary development applications

Other OpenStep frameworks

Useful command-line tools

Other programming resources



Appendix B

Programming Tools and Resources

There is more to the Apple development environment than just Project Builder and Interface Builder. This appendix describes some of the other applications, frameworks, command-line utilities, and other resources that are available to programmers.

Other Development Applications

The Apple development environment includes applications other than Project Builder and Interface Builder. Except where noted, these applications are installed in **/System/Developer/Apps**.

Name	Description
FileMerge	Visually compares the contents of two files or two directories. You can use FileMerge, for example, to determine the differences between versions of the same source code file or between two project directories. You can also use it to merge changes.
MallocDebug	Measures the dynamic-memory usage of applications, finds memory leaks, analyzes all allocated memory in an application, and measures the memory allocated since a given time.
Yap	A utility for editing and previewing PostScript code.
Sampler	Analyzes performance problems with your application by sampling the call stack of your program over a period. (In /System/Developer/Demos)

Other Installed Frameworks

A framework contains a dynamic shared library, related header files, and resources (including nib files, images, sounds, documentation, and localized strings) used by the library. All frameworks are installed in **/System/Library/Frameworks**. The Apple development environment provides these other frameworks besides the Application Kit, Display PostScript, and the Foundation frameworks:

Name	Description
System	Operating-system and low-level Objective-C run time APIs
SoundKit	Sound recording, playback, and editing capabilities.
InterfaceBuilder	Creation of custom static (compiled) palettes for use in Interface Builder
ProjectBuilder	Creation of custom project types, source-code management (SCM) adaptors, and other Project Builder extension bundles.

Useful Command-Line Tools

Apple has created or modified several tools for compilation, debugging, performance analysis, and so on. The following table lists some of the more useful of these tools. You can get further information using the man pages system.

Name	Description	Location
cc	Compiles C, Objective-C, C++, and Objective-C++ source code files.	/bin
gdb	Source-level symbolic debugger for C, extended by Apple to support Objective-C, C++, Mach, Windows NT, and (by late 1996) Windows 95.	/bin
gnumake	Utility for making programming projects.	/bin
as	Assembler; translates assembly code into object code.	/bin
defaults	Reads, writes, searches, and deletes user defaults. The defaults system records user preferences that persist when the application isn't running. When users specify defaults in an application's Preferences panel, NSUserDefaults methods are used to write the defaults.	/usr/bin
pswrap	Creates C functions that "wrap" PostScript code and send it to the Window Server for interpretation.	/usr/bin
nibTool	Reads the contents of an Interface Builder nib file. Prints classes, the hierarchy, objects, connections, and localizable strings.	/usr/bin
libtool	Creates static or dynamic libraries from specified object bin files for one or multiple architectures.	
otool	Displays specified parts of object files or libraries.	/bin
nm	Displays the symbol table, in whole or in part, of the specified object file or files.	/bin
AnalyzeAllocation	Analyzes program memory allocation.	/usr/bin
fixPrecomps	Creates or refreshes a precompiled header file for each of the major frameworks.	/usr/bin
strip	Removes or modifies the symbol table attached to assembled and linked output.	/bin
lipo	Creates, lists, and manipulates multi-architecture object files	/bin

Other Programming Resources

You can find programming resources—such as fonts, sounds, and palettes—in various subdirectories of **/System/Library**.

Name	Comments
SystemResources	Character-set information and location of headers for automatic precompilation (fixPrecomps)
Colors	Bundles containing the default set of color binaries for the Colors panel
Fonts	Default set of system fonts, including AFM, bitmap, and outline versions
PS2Resources	PostScript files containing calibrated color space and color rendering, printing halftones, and gray-shading patterns
Rulebooks	Glyph generators for various string encodings
Sounds	Default sound files (“ .snd ”) such as Cricket, Ping, and Rooster

Normally these resources are accessed via Application Kit APIs. Be careful about having dependencies on these resources in your code since the location and format of these resources might change in future releases.