;SettingCompilerAndLinkerOptions.rtfd;;¬ Previous Section

Customizingyourmakefiles;¬Customizing your makefiles

arrow.eps ¬ Set information in the Build Attributes inspector.

Or

arrow.eps ¬ Edit the files Makefile.preamble and Makefile.postamble.

Sometimes it's necessary to alter the standard build process as defined by the project makefile. If you need to do this, first look for the options you need to change in the Build Attributes inspector. Project Builder uses the information you set in the Build Attributes inspector to index the project. So to make sure the project index is correct, you should use this inspector instead of editing the makefile directly. (Project Builder updates the makefile for you.)

BuildInspectorCompilerFlags2.eps ¬

If the inspector does not have an option for what you need to set, edit the files **Makefile.preamble** and **Makefile.postamble**. Both files contain makefile variable definitions, and the comments in these files describe what each macro defines.

In general, **Makefile.preamble** contains macros that add to the standard makefile definitions, and **Makefile.postamble** contains macros that override the standard definitions. For example, the **LIBS** macro defines the libraries that your program should link with. The standard makefile sets this macro to the libraries in the project. If you want to change this definition, you uncomment the **LIBS** definition in **Makefile.postamble** and change the definition. However, if you want to link with more libraries than those added to the project, set **OTHER_LIBS** in **Makefile.preamble** to the additional library's name.

ªCreating your own build targets,º ;CreatingYourOwnBuildTargets.rtfd;;¬ ªSetting search paths,º ;SettingSearchPaths.rtfd;;¬ and ªSetting compiler and linker optionsº ;SettingCompilerAndLinkerOptions.rtfd;;¬

in this chapter describe some of the fields in the Build Attributes inspector.

If you're building a library or framework, see Chapter€12, ªCreating Frameworks and Dynamic Shared Libraries,º for more information on setting up the makefiles.
;../../05_SpecialTasks/12_FrameworksLibraries/FrameworksLibrariesConcepts.rtfd;MacrosfortheMakefileHacker;¬

Reducing Compile Time

Each build begins by exporting any public headers to a location where they are visible to the rest of the project (for example, headers in subprojects are exported to the **derived_src** directory if you mark them as Project Headers in the File Attributes inspector). If you're building a project that does not export any headers (no boxes are turned on for header files in the File Attributes inspector), such as an application or tool with no distributed objects or library API, you can omit this step by setting this macro in **Makefile.preamble**:

Preamble Macro TableHeadRule.eps ¬	**Description**
SKIP_EXPORTING_HEADERS TableRule.eps ¬	Skips the exporting headers step of the build.

Adding Make Dependencies

If you add dependencies or targets to the makefile, set these macros in **Makefile.preamble**.

Preamble Macro TableHeadRule.eps ¬	**Description**
OTHER_PRODUCT_DEPENDS TableRule.eps ¬	Dependencies you defined that should be used in all builds.
OTHER_INITIAL_TARGETS TableRule.eps ¬	Targets you defined that should be built before subprojects.
OTHER_INSTALL_DEPENDS TableRule.eps ¬	Dependencies you defined that should be used for the install target.

Setting Up The Install Target

To set up the install target to work the way you want, set these macros in **Makefile.preamble**.

Preamble Macro TableHeadRule.eps ¬	**Description**
DSTROOT TableRule.eps ¬	Path to prepend to the installation path specified in the Build Attributes inspector. The default is ***/***.

And these macros in **Makefile.postamble**.

Postamble Macro TableHeadRule.eps ¬	**Description**
INSTALL_AS_USER TableRule.eps ¬	The owner of the installed files. The default is root.
INSTALL_AS_GROUP TableRule.eps ¬	The group for the installed files. The default is wheel.
INSTALL_PERMISSIONS TableRule.eps ¬	The installed files' permissions. The default is read and execute permissions turned on for all users.
APP_STRIP_OPTS TableRule.eps ¬	Stripping options to pass to the **strip** tool. The default is no options, which strips debugging symbols out of the executable. Only set options here if the application loads other bundles.
APP_WRAPPER_EXTENSION TableRule.eps ¬	The extension to use for application's output. The default is **.app**.

Setting Up Make Clean

To set up **make clean** to work the way you want, set these macros in **Makefile.preamble**.

Preamble Macro	**Description**
TableHeadRule.eps ¬	
OTHER_GARBAGE	Files that should be deleted in addition to object files and executables.
TableRule.eps ¬	
CLEAN_ALL_SUBPROJECTS	If defined, **make clean** cleans subprojects as well. This macro is defined by default. To undefine it, comment it out in **Makefile.preamble**.
TableRule.eps ¬	

Overriding Compiler and Linker Options

Most targets produce an optimized, debuggable executable and do not suppress compiler warnings. To change the compiler options used to produce the usual executable, override these macros in **Makefile.postamble**.

Postamble Macro	**Description**
TableHeadRule.eps ¬	
OPTIMIZATION_CFLAG	Compiler optimization option, used by all but the debug target. The default is **-O**, which reduces code size and execution time.
TableRule.eps ¬	
LOCAL_DIR_INCLUDE_DIRECTIVE	Override if you don't want the current directory in the default search path. By default, this is defined as **-I.**
TableRule.eps ¬	
DEBUG_SYMBOLS_CFLAG	Compiler debug symbols options, used by all but the install target. The default is **-g**, which produces line number and symbol information.
TableRule.eps ¬	
WARNING_CFLAGS	Compiler warning message level, used by all targets. The default is **-Wall**, which suppresses none of the warning messages.
TableRule.eps ¬	
DEBUG_BUILD_CFLAGS	Compiler options used only by the debug target. The default is **-DDEBUG**, which defines

	the **DEBUG** preprocessor macro.
TableRule.eps ¬	
PROFILE_BUILD_CFLAGS	Compiler options used only by the profile target. The default is **-pg**, which produces information for **gprof**, and **-DPROFILE**, which defines the **PROFILE** preprocessor macro.
TableRule.eps ¬	

Setting Up Other Tools

For more information about the tools listed here, see their man pages.

Some tools are invoked by **make** if the project contains files with certain extensions. To set up these tools, set these macros in **Makefile.preamble**.

Preamble Macro	**Description**
TableHeadRule.eps ¬	
PSWFLAGS	Options for the **pswrap** tool (invoked on **.psw** files).
TableRule.eps ¬	
YFLAGS	Options for the **yacc** tool (invoked on **.y.c** or **.ym.m** files).
TableRule.eps ¬	
LFLAGS	Options for the **lex** tool (invoked on **.l.c** or **.lm.m** files).
TableRule.eps ¬	
MSGFILES	Input files for the **msgwrap** tool. These should have the **.msg** extension.
TableRule.eps ¬	
DEFSFILES	Input files with a **.defs** extension for the **mig** tool.
TableRule.eps ¬	
MIGFILES	Input files with a **.mig** extension for the **mig** tool.
TableRule.eps ¬	
RPCFILES	Input files for the **rpcgen** tool (invoked on **.x** files).
TableRule.eps ¬	

Including More Files in the Build

There may be files that you don't want to add to the project but that should be included in the build. Use these macros in **Makefile.preamble** to have the build handle more files.

Preamble Macro TableHeadRule.eps ¬	**Description**
OTHER_LIBS TableRule.eps ¬	Libraries to link with besides the libraries included in the project.
OTHER_OFILES TableRule.eps ¬	Object files to link into the executable besides those produced by the source files in the project.
OTHER_SOURCEFILES TableRule.eps ¬	Source files besides those included in the project.
INCLUDED_ARCHS TableRule.eps ¬	Architectures to which this project or subproject should be restricted to building for. Building for other architectures is skipped. Must be a subset of the architectures selected in the Build Options panel.
EXCLUDED_ARCHS TableRule.eps ¬	Similar to INCLUDED_ARCHS, but lists architectures that this project or subproject shouldn't build for instead of architectures it should build for. Don't use if using INCLUDED_ARCHS.