Appendix: The Static Analyzer

Use the static analyzer to find bugs in your code before you even run your app. The static analyzer tries out thousands of possible code paths in a few seconds, reporting potential bugs that might have remained hidden or bugs that might be nearly impossible to replicate. This process also identifies areas in your code that don't follow recommended API usage, such as Foundation, UIKit, and AppKit idioms.

To perform static code analysis, choose Product > Analyze. The Xcode static analyzer parses the project source code and identifies these types of problems:

- Logic flaws, such as accessing uninitialized variables and dereferencing null pointers
- Memory management flaws, such as leaking allocated memory
- Dead store (unused variable) flaws
- API usage flaws that result from not following the policies required by the frameworks and libraries the project is using

The static analyzer reports problems in the issue navigator, available by clicking the Issue Navigator button in the project navigator bar. Select an analyzer message in the issue navigator to display the associated code in the source editor. Click the corresponding message in the source editor. Use the pop-up menu in the analysis results bar above the source code editor to study the flow path of the flaw. Then edit the code to fix the flaw.

Performing Static Code Analysis

Find flaws—potential bugs—in the source code of a project with the static analyzer built into Xcode. Source code may have subtle errors that slip by the compiler and manifest themselves only at runtime, when they could be difficult to identify and fix.

To find flaws in your source code using the static analyzer:

1. Choose Product > Analyze.
2. In the issue navigator, select an analyzer message.
3. In the source editor, click the corresponding message.
4. Use the pop-up menu in the analysis results bar above the edit area to study the flow path of the flaw.
5. Edit the code to fix the flaw.

You can suppress false positive messages from the analyzer using assertions, attributes, or pragma directives.

When you analyze a project for the first time, you may uncover a lot of issues. But if you run the static analyzer regularly and fix the flaws it uncovers, you should see fewer problems in subsequent analyses. Analyze early; analyze often. It's good for the code.

> **Note:** if the static analyzer reports no problems, you can't assume that there are none. The tool cannot necessarily detect all the flaws in the source code.

> 💡 **Tip:** Choose Product > Clean to remove the analyzer messages from the issue navigator.