

Adding public API;↵Adding public API

- 1 **Change the compatibility project version number and the current project version number in Makefile.postamble.**
- 2 **Build the project.**

You shouldn't change the major version number when you add API (for example, if you add a class, add a method to an existing class, or add a function or constant). Adding API doesn't break existing programs. Existing programs are guaranteed to be using the older API and will still run because you've left the older API intact. However, new programs might use the new API, and therefore shouldn't try to run against older versions of the framework, which don't define that API.

VersionMacros.eps ↵

When you add API, increment the compatibility version number. The compatibility version number protects programs linked with newer versions of a library from running with older versions of the library. In order for a program to launch, the compatibility version number of the framework it runs with must be equal to or greater than the **CURRENT_PROJECT_VERSION** number of the framework it linked with.

AddPubAPI.eps ↵

Why shouldn't you just change the major version number when you add API? Because programs linked with the previous version of the framework still run with the new version. If you change the major version number, the previous version remains installed on your users' systems. By changing the compatibility version number instead, you can install just one version.

Remember that adding instance variables to a class is an incompatible change, which means you should change the version name instead of the compatibility version number. See ^aTips and Tricks to Changing the Major Version^o for an explanation. [;FrameworksLibrariesConcepts.rtf](#);TipsandTrickstoChangingtheMajorVersion;↵

Increment the value in **CURRENT_PROJECT_VERSION** whenever you change the compatibility version number. See ^aCURRENT_PROJECT_VERSION: For That Extra Level of Checking^o in this chapter.

;FrameworksLibrariesConcepts.rtfd;CURRENT_PROJECT_VERSION:ForThatExtraLevelofChecking;~