

# Naminganewclass;↵Naming a new class

- 1 In Interface Builder, display the Classes display of the nib file window.
- 2 Select the class you want your class to inherit from.
- 3 Choose Subclass from the Operations menu.
- 4 Type the name of your class over the highlighted <sup>a</sup>default<sup>o</sup> name.

When you create an application, you must create at least one subclass to do anything meaningful. The OpenStep frameworks do a lot of the work for you, but you must always supply, in one or more subclasses, the distinctive logical and computational flow of your application.

\_NamingANewClass1.eps ↵

When you subclass, the first thing you must do is select your class's superclass. Ideally, the superclass of your class should behave much the way you want your class to behave. Your class merely adds the behavior you want to what the superclass offers, or modifies the superclass's behavior in some way. Often the behavior you want is so bound to resolving a particular problem that the proper choice of superclass is NSObject because it provides the most generic behavior.

\_NamingANewClass2.eps ↵

**Tip:** Pressing Return when a class is selected is equivalent to choosing the Subclass command.

The new class is listed under its superclass with a default name: the superclass name prefixed with <sup>a</sup>My<sup>o</sup> (such as <sup>a</sup>MyNSObject<sup>o</sup>). Replace this default name with the new name.

\_NamingANewClass3.eps ↵

Later, if you want to rename the class, first re-select the class name by double-clicking it. Then type the new name, replacing the selected text.

See <sup>a</sup>[A Short Practical Guide to Subclassing](#)<sup>o</sup> in this chapter for more on the relation between superclasses and subclasses. [;SubclassingConcepts.rtf;AShortPracticalGuidetoSubclassing;-](#)

**Related Concept:** [;SubclassingConcepts.rtf;APerspectiveontheClassHierarchy;-](#) A Perspective on the Class Hierarchy