

Weak Linking Support

Applications that link against iOS SDK 4.2 or later can now take advantage of new support for determining the availability of Objective-C classes. System frameworks now tag their classes with availability information that the compiler can use to link weakly to those classes as needed. This new tagging mechanism simplifies the code you need to use to do runtime checking for the availability of the class. Whereas previously, you needed to use the [NSClassFromString](#) function to determine whether a class was present, now you can simply check for the existence the class directly using code similar to the following:

```
if ([UIPrintInteractionController class])
{
    // Create an instance of the class and use it.
}
else
{
    // The print interaction controller is not available.
}
```

In order to use this feature, you must do the following:

- Build your project using LLVM and Clang. (The standalone GCC compiler does not currently support this feature.)
- Set your project's deployment target to iOS 3.1 or later.
- Change the linking option for each framework you want to link weakly. To do this, open an inspector for your application target and go to the inspector's General tab to view the list of frameworks and libraries linked against the target. For frameworks you want to link weakly, change the value in the Type column from Required to Weak.

If the deployment target for your application is set to a version of iOS that does not contain a given framework (because it was introduced in a later version), set the link type of that framework to Weak. For most other frameworks, you should leave the link type set to Required.

For information about how to perform runtime availability checks for classes and methods, see [SDK Compatibility Guide](#).