

Q: How can I get at the **protected** or **private** instance variables of a class?

A: There is a construct which allows you to access all of the instance variables of an object as if it were a C-structure. We don't encourage it (although it is documented in the Objective-C chapter of the Concepts manual). But if you really really must, then you can use the `@defs` construct. `@defs` allows you to essentially import all the instance variables of a given class into a structure local to your application.

```
struct InternalDetails
{
    @defs(Matrix);
};
```

```
struct InternalDetails *matrixGoodies;
```

```
matrixGoodies = (struct InternalDetails *)[Matrix new];
if (matrixGoodies->mFlags.radioMode == ...)
```

**Important Note:** Never use this technique to set or look at the values of instance variables whose names start with an underline in the Application Kit. The contents of those variables are for

internal use and subject to change with releases. It's OK to use

```
@defs or Matrix *m;
```

type constructs to get at the **private** or **protected** variables which don't begin with an underline. However you should only do this as a last resort.

*TERMINOLOGY NOTE: Prior to the Release 3, there were only **public** and **private** instance variables. By default, all instance variables were **private** unless specified as **public**. Release 3 introduced the concept of **protected** instance variables. **Protected** instance variables are available to a class and its subclasses. **Private** instance variables are available to that class only—not to the subclasses. What was a **private** instance variable under Release 2 is a **protected** variable under Release 3. By default all instance variables under Release 3 are **protected**.*

*See the Release 3 Compiler release notes for further information on **private/protected/public** instance variables.*

QA605

Valid for 1.0, 2.0, 3.0