



```
//MARK: -
func check() -> Bool
{
    var myCode : SecCode?
    var status : OSStatus?
    var requirement : SecRequirement?
    var errors : UnmanagedCStrings?

    let entitlement = "anchor apple generic and identifier = \"com.yourapp.example\" and certificate leaf[subject.OU] = \"YOURTEAMID\""
    status = SecRequirementCreateWithStringAndErrors(entitlement as CFString, [], kErrors, kRequirement)
    if status != errSecSuccess
    {
        NSLog("SecRequirements failed with status code: %(status).")
        // optionally unpack kErrors here
    } else
    {
        status = SecCodeCopySelf([], myCode)
        if status != errSecSuccess
        {
            NSLog("SecCodeCopySelf failed with status code: %(status).")
        } else
        {
            status = SecCodeCheckValidity(myCode, [], requirement)
            if status != errSecSuccess
            {
                NSLog("SecCodeCheckValidity failed with status code: %(status).")
            } else
            {
                NSLog("Codesigning: valid")
                return true
            }
        }
    }
}
```

f[subject.OU] = "YOURTEAMID"

f[subject.OU] = "YOURTEAMID"

```
        NSLog("SecCodeCheckValidity failed with status code: %(status).")
    } else
    {
        NSLog("Codesigning: valid")
        return true
    }
}
```