
Call Of Duty Black Ops 2 English Language Pack.rar

.Q: Can weak equality be used in conjunction with a variable in a closure? I am trying to get Swift to translate a string literal in a closure, and have come across a Swift syntax I had not seen before. Specifically, I am invoking a function which itself wraps a closure and which was written in C++. A line like this: tableView.delegate = self { tableView in print("TableView delegate called") } will compile but fail at runtime with an exception: fatal error: unexpectedly found nil while unwrapping an Optional value If I change the code to: tableView.delegate = self { tableView in print("TableView delegate called") return true } the code compiles and the delegate gets called as expected. I am not sure what to make of this - can anyone explain what is going on? Here is a full example demonstrating the issue. I hope I'm not breaking any rules by posting this - in case I am, I'd like to know so I can fix it. class CppTableDelegate: UITableViewDelegate { func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat { return 200 } func tableView(_ tableView: UITableView, heightForHeaderInSection section: Int) -> CGFloat { return CGFloat(3) } func tableView(_ tableView: UITableView, viewForHeaderInSection section: Int) -> UIView? { return UIView() } func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell { return UITableViewCell() } func tableView(_ tableView: UITableView, didSelectRowAt indexPath

[Download](#)

