<http://www.netobjectivestest.com/PatternRepository/index.php?title=AdapterVersusProxyVersusFacadePatternComparison>

# Adapter versus Proxy Pattern Comparison

One of the more frequent questions I get in class is "what's the difference between [Adapter](http://www.netobjectivestest.com/PatternRepository/index.php?title=TheAdapterPattern) and [Proxy](http://www.netobjectivestest.com/PatternRepository/index.php?title=TheProxyPattern)? They really seem the same to me". This is mostly because the runtime relationships are awfully similar:

[](http://www.netobjectivestest.com/PatternRepository/index.php?title=Image:AdapterProxyFacade.jpg)

This is a good example of the critical notion that patterns are not diagrams, or code snippets.

## Proxy vs. Adapter

* The [Proxy](http://www.netobjectivestest.com/PatternRepository/index.php?title=TheProxyPattern) changes the behavior of the Service, but preserves its interface.
* The [Adapter](http://www.netobjectivestest.com/PatternRepository/index.php?title=TheAdapterPattern) changes the interface of the Service, but preserves it behavior.
* A Client can use the [Proxy](http://www.netobjectivestest.com/PatternRepository/index.php?title=TheProxyPattern) or the Service Entity in the same way.
* A Client designed to use the [Adapter](http://www.netobjectivestest.com/PatternRepository/index.php?title=TheAdapterPattern) would not be able to use the Service Entity without it.
* The [Proxy](http://www.netobjectivestest.com/PatternRepository/index.php?title=TheProxyPattern) can be cast to the interface of the Service. The [Adapter](http://www.netobjectivestest.com/PatternRepository/index.php?title=TheAdapterPattern) can be cast to the interface the Client expects.