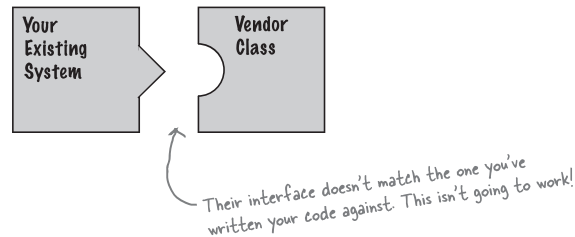
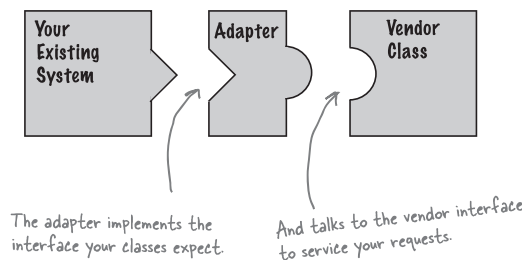


*the adapter pattern***Object oriented adapters**

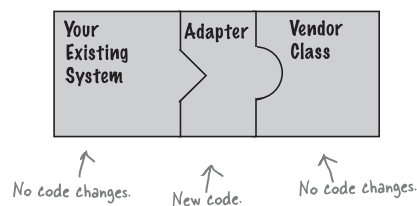
Say you've got an existing software system that you need to work a new vendor class library into, but the new vendor designed their interfaces differently than the last vendor:



Okay, you don't want to solve the problem by changing your existing code (and you can't change the vendor's code). So what do you do? Well, you can write a class that adapts the new vendor interface into the one you're expecting.



The adapter acts as the middleman by receiving requests from the client and converting them into requests that make sense on the vendor classes.



Can you think of a solution that doesn't require YOU to write ANY additional code to integrate the new vendor classes? How about making the vendor supply the adapter class.

you are here ▶ 237

**Chapter 7. Being Adaptive**

Head First Design Patterns By Eric Freeman, Elisabeth Freeman, Bert Bates, Kathy Sierra  
 ISBN: 0596007124 Publisher: O'Reilly  
 Print Publication Date: 2004/10/25

Prepared for Ann Cherkis, Safari ID: maottw@gmail.com

User number: 1673621 Copyright 2008, Safari Books Online, LLC.

This PDF is exclusively for your use in accordance with the Safari Terms of Service. No part of it may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates the Safari Terms of Service is strictly prohibited.