

PHP Tutorial 5

XML and XSLT

Getting News Off the Web

In this tutorial, you'll learn how to create an XSLT fragment that formats news headlines from the *National Exasperator*. The headlines are contained in an XML file. You'll add the XSLT fragment to a dynamic page on the same site you've been working on in tutorials throughout this book. To get started, you need to grab the dynamic site files from www.sawmac.com/dw8/php/, and you'll have to have set up a Web server and application server as described in the first PHP tutorial. However, you don't need to have finished any of the other PHP tutorials to do this one.

Note: You'll need to have XSLT support with your installation of PHP. If you used the XAMPP package discussed in the first tutorial, it's built-in so you needn't worry. To determine if XSLT support is enabled, create a page called `info.php` with this one line of code in it `<?php phpinfo(); ?>`. Load the Web page from the Web server in your Web browser—a page listing all sorts of info about your Web server and PHP appears. Search the page for this “XSLT Support” and look to see if the word “enabled” is next to it (this might also appear as “EXSLT Support”). If so, you're golden, if not, you'll need to reinstall PHP with XML support—unfortunately, most Web hosting companies won't do this for you without a lot of nagging, so if it's not enabled on your Web hosts server, dash off an e-mail to their customer service department requesting that XSLT support be enabled.

If you put the `info.php` page on your Web hosting server (the one connected live to the internet) remove it. The information presented by this page can be extremely valuable to hackers trying to figure out how to make your life miserable.

Start by creating a new XSLT fragment.

1. Choose File→New.

The New Document window appears.

2. Make sure the General tab is selected. Choose Basic Page from the category list, and XSLT (fragment) from the basic page list. Click Create.

The Locate XML Source window appears (see Figure 24-1 on page 848 of the printed book), from which you can tell Dreamweaver which XML file to use. This can be either the XML file located with the tutorial files (*feed.xml*), or, if you're connected to the Internet try the online version of the file

Copyright 2005, Sawyer McFarland Media, Inc. This material is supplied to readers of *Dreamweaver 8: The Missing Manual* as a helpful supplement to that book. If you don't own the book, what are you doing reading this? Please go out and buy the book: it's a good book, explains a lot more than this tutorial, and you'll feel better about yourself knowing you did the right thing. Thanks.

located on the National Exasperator Web site at <http://www.nationalexasperator.com/feed.xml>. Using the online file lets you simulate what it's like to access an RSS feed from a news site like CNN or Wired.

- To use the local XML file, click the Browse button, navigate to the *news* folder in the site's root folder, and then select the *feed.xml* file. To use an XML file from some online source, select the "Attach a remote file on the Internet" button, and then type <http://www.nationalexasperator.com/feed.xml>.

In this particular case, both files are identical, so after this step, everything else in the tutorial will be the same.

- Whichever method you use, click the OK button to close the Locate XML Source window.

If you have the Bindings panel open (Window→Panel), you'll see it fill with the tags from the XML file (see Figure PHP Tutorial 5-1).

- Choose File→Save, and save this file as *news.xml* in the *news* folder.

You've just created an XSLT stylesheet! Next you'll add XML data to it.

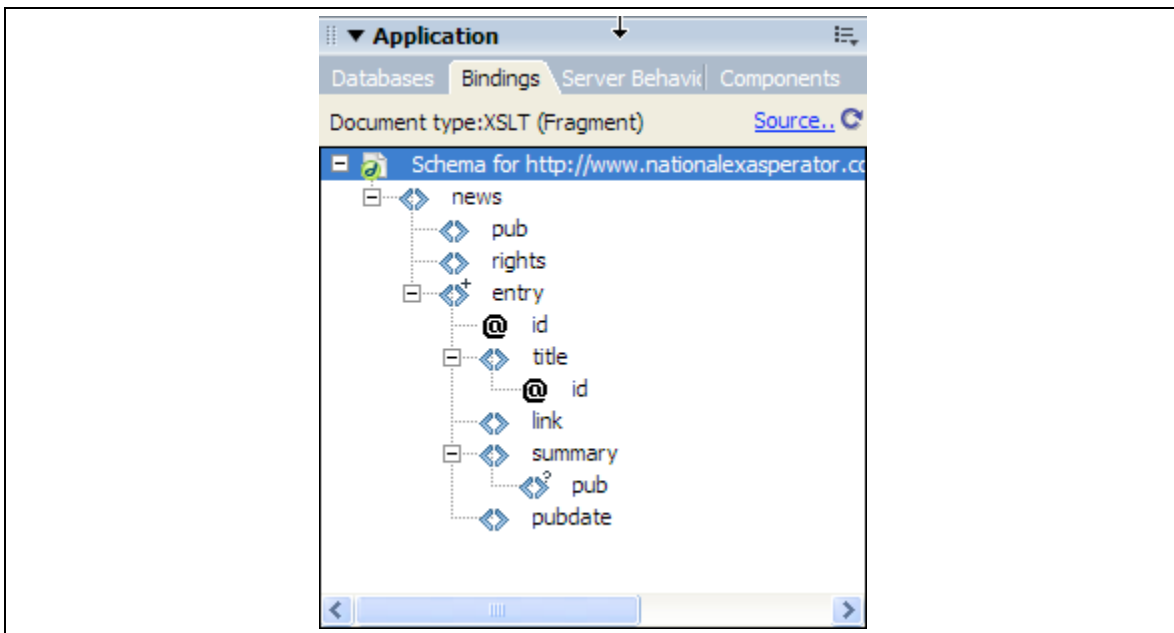


Figure PHP Tutorial 5-1. When working with an XML file, the Bindings panel displays all of the XML tags contained in the file. See page 855 for more on how the Bindings panel works.

- Drag the *title* tag from the Bindings panel into the Document window.

You'll see `{news/entry/title}` with a blue background on the page. The blue background indicates that this is dynamic data—just as you see when adding recordset information to database-driven pages. The "news/entry/title" is an XPath expression, which, translated, means: use the "title" tag, which is located inside the "entry" tag, which is located inside the "news" tag. (The "news" tag is the "root" or top-level element in the XML file.) The XPath expression is much like the nesting of HTML tags that you see listed in the Document window's Tag Selector (see page 21 in the book.)

- In the Property Inspector, choose Heading 2 from the Format menu.

Alternatively, you can use the keyboard shortcut Ctrl-2 (@cmd-2) to format the newly inserted chunk of XML as a header. One problem with designing just a fragment of a larger page is that you don't really know what the design will look like once it's included in your final dynamic Web page.

Remember, an XSLT fragment will appear as just *one* chunk of code within the larger canvas of a dynamic Web page. The page probably will use its own set of CSS styles—so a Heading 3 on that page may already be formatted with a particular style.

Fortunately, Dreamweaver includes a great feature that lets you design an XSLT fragment using the styles from the dynamic page on which the fragment will ultimately be displayed: it's called Design Time Style Sheets. (Yep, this feature works just like the Design Time feature you learned about on page 174.)

8. Choose Text→CSS Styles→Design Time...

The Design Time Style Sheets window appears. Here you'll tell Dreamweaver which external stylesheet to use while you design the page.

9. Click the + button. In the Window that appears, navigate to the local root folder of the site, select the *global.css* file, and then click OK.

The file is now listed in the Design Time Style Sheets window, which means that you can apply any of the already-created styles from the site to the XML data on this page.

10. Click OK to close the Design Time Style Sheets window.

National Exasperator's distinctive grey sidebar strip appears in the background of the page. But there are a couple of things wrong here. First, the headline overlaps the sidebar; on the final page it will actually go inside a table cell that holds the page's main content. Second, the Heading 2 isn't displaying as it will in the final page, since there it will be inside a table cell that has a CSS class style named *mainContent* applied to it. A Heading 2 (or h2 tag) looks differently when placed inside that class style thanks to the power of "Descendent Selectors" (see page 189).

At any rate, you need to take a couple of extra steps to make it easier to use Dreamweaver's visual design tools (thereby saving yourself from having to muck around in Code view). First, you need to wrap the h2 tag in a temporary <div> tag.

11. In the Document Window, click on the XML tag you added in step 6, and then choose Insert→Layout Objects→Div Tag.

The Insert Div Tag window appears.

12. Make sure "Wrap Around Selection" is selected in the first menu, and select *mainContent* from the Class menu. Click OK.

The design suddenly changes. The newly added *mainContent* div tag shows how the fragment will ultimately look. Since the tag will ultimately be placed inside a table cell with the class *mainContent* applied, you're now seeing the tag's true "context" within the page. In addition, because of a descendent selector that specially formats h2 tags that appear inside any tag with the class *mainContent* applied, the heading also changes appearance.

As brain-churning as all these details can sometimes seem, you'll find that using Design Time Style Sheets will make designing XSLT fragments a lot easier.

13. Click to the right of the XML data ({news/entry/title}) and press the space key.

You'll add the date the headline was published next.

14. Drag the *pubdate* tag from the Bindings panel to the right of {news/entry/title} to add it to the page. From the Style menu in the Property Inspector, choose "date."

You can apply CSS styles to any of the XML data you add to a page. In this case, the style is from the Design Time Style Sheet you added earlier.

15. Click to the right of the *pubDate* tag, and press Enter (Return) to create a new paragraph.

You're almost done adding information from the XML file. You'll add the story summary, and then the *National Exasperator* copyright notice.

16. Drag the summary tag from the Bindings panel to the new paragraph.
And, finally, you'll add the copyright notice.
17. Click to the right of the *summary* tag, press Enter (Return), and then drag the *rights* tag from the Binding panel to the page. In the Property Inspector, choose "copyright" from the Style menu.
You're done with the basics. You've added XML and all the formatting you'll need to make the page look pretty. Next you'll learn how to add a link from an XML file.

Adding a Repeat Region and a Link

The XSLT fragment you created in the previous section is nearly complete. However, if you used it as is, only one headline, publication date, and summary would be displayed. Many XML files include multiple versions of the same set of tags (just like a recordset from a database often includes multiple records.) The XML file you're using actually lists several headlines (the *National Exasperator* is a busy publication). To display multiple sets of the same tags from an XML file, you need to use a Repeat Region.

1. In the Document window, click to the right of the story summary (`{/news/entry/summary}`) and drag up and to the left until you've selected the summary, publication date, and title.
You'll turn this selection into a repeat region, so that all of this information will be displayed one time for each story in the XML file.
2. Choose Insert→XSLT Objects→Repeat region.
The XPath Expression Builder window appears (see Figure PHP Tutorial 5-2). You simply have to select the tag that represents a story in the XML file—in this case, the tag's name is *entry*.
3. Select the *entry* tag, and then click OK.

A grey box appears around the selection and a grey tab with the label "xsl:for-each" appears in the top left corner of the box. The box represents the repeating region. In addition, you'll notice that those long-winded names—`{/news/entry/summary}`, and so on—have changed to shorter and simpler labels: title, pubdate, and summary.

The headline and news summary are just teasers for the real content. Fortunately, the XML file you're using contains a link to a Web page with the juicy story behind the headlines. Linking headlines and summaries to full stories is a common practice with feeds from news Web sites and blogs. The feed usually contains just a brief description of the story or blog posting, and then a link pointing to the whole story. Next, you'll add a link to the headlines.

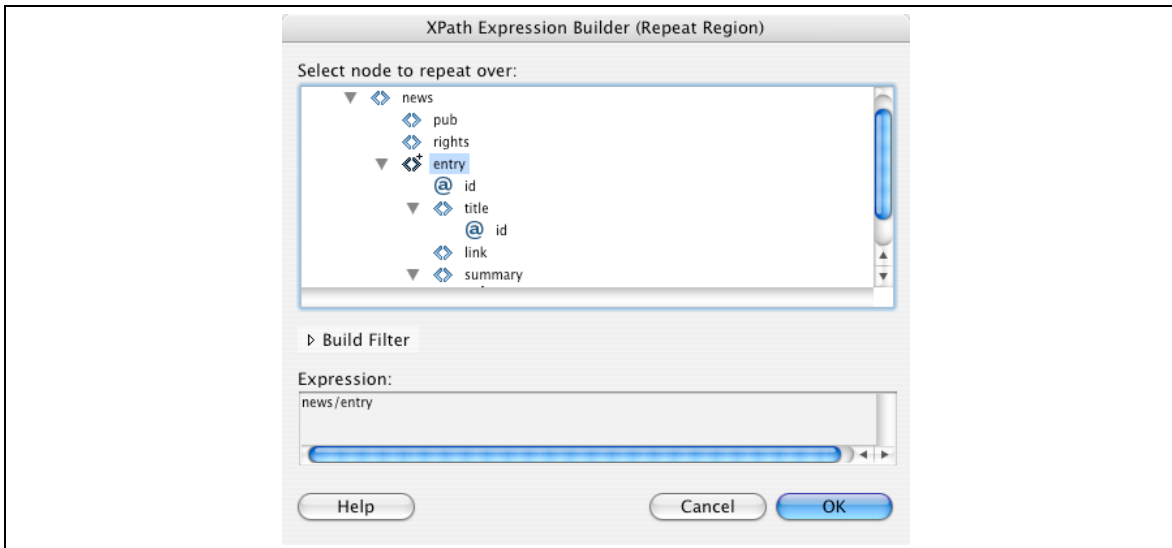


Figure PHP Tutorial 5-2. To repeat a region, you must select a tag that appears multiple times, and which contains the tags you want to repeat. The + sign next to a tag icon indicates that the tag repeats multiple times in the XML file. In this example, you can see that the tags you want to repeat—title and summary, for example—are nested inside the entry tag.

4. In the Document window, click the XML tag labeled {title} to select it.

You'll add a link to this tag.

5. In the Property Inspector, click the browse-for-file folder button.

The Select File window opens. (For a refresher on creating links see page 108.)

6. Click the Data Source radio button.

The tags from the XML file appear in the Select File window (see Figure PHP Tutorial 5-3). Most of these tags wouldn't make sense as a link, since they contain just text or other tags, but the XML file happens to include a URL to a Web page in the <link> tag.

7. Select the link tag, and then click the OK button.

Dreamweaver turns the headline into a link.

Believe it or not, that's all there is to creating the XSLT file. You need to do one last thing to get the XSLT file ready to insert into a dynamic page.

Note: You need to add the Repeat Region *before* the link, due to a bug in Dreamweaver 8. If you add the link first, and then select all of the text you wish to repeat (as in step 1 above), Dreamweaver doesn't actually select the entire link tag. The link then won't work after you convert this partial selection to a repeat region. This problem only happens when the link is added to the first part of the information you wish to repeat, and then again only in some unusual circumstances. But since you might run into this nuisance, it's best to add the repeat region first, and then the link.

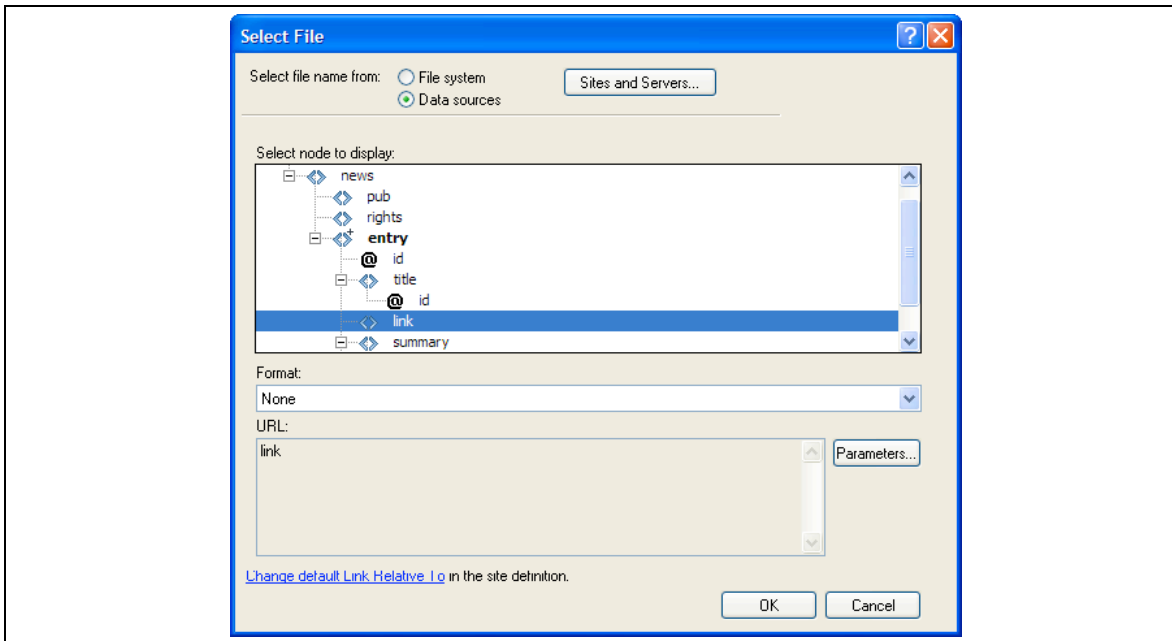


Figure PHP Tutorial 5-3. The “Select node to display” box let’s you use the content of an XML tag as a link. In XML-speak a node is one of the discrete parts of an XML document, such as a tag, a property, a comment, or even just plain text. In this case, node refers to either the contents of the tags, or the contents of the attribute of a tag.

8. In the Tag Selector (in the bottom-left corner of the Document window), right-click (Control-click) the tag `<div.mainContent>`. From the shortcut menu that appears, select Remove Tag.

Remember you placed that `<div>` tag temporarily in the file so you could accurately format the XML data using Cascading Style Sheets? You no longer need that tag, since the design is done and the dynamic page you’re adding this XSLT fragment to already has that tag in place.

9. Save and close the XSL file.

Adding the XSLT Fragment to a Dynamic Page

The last piece of the puzzle is adding the XSLT fragment to a dynamic page using a Dreamweaver Server Behavior. This procedure adds the necessary programming code to magically transform the XML, XSLT, and so on, into browser-readable HTML.

1. Open the page `index.php` located in the `news` folder in the root folder of the site.
User your favorite technique: For example, choose File→Open and select the file, or open the Files panel, and double-click the file to open it.
2. Click in the empty space directly below the headline “Latest Headlines”.
This spot is where you’ll place the XSLT fragment, but due to a small bug in Dreamweaver 8, you need to do one thing first.
3. In the Property Inspector choose “None” from the Format menu.

The empty space was actually an empty paragraph complete with the HTML `<p>` tags. Changing the format to “none” removes the `<p>` tags, which is a good thing. Otherwise, all the HTML from the XSLT style sheet would be placed *inside* of this `<p>` tag, meaning you’d have `<h2>` and other `<p>` tags nested inside of the paragraph. That arrangement would not only make the page look weird when viewed in a browser, but it’s also invalid HTML—shame on you, Dreamweaver.

4. Open the Server Behaviors panel (Window→Server Behaviors.) Click the + button and then select XSL Transformation.

You can also click the XSLT button on the Application tab of the Insert bar—it's the last button on the right of the tab. The XSL Transformation window opens (see Figure PHP Tutorial 5-4). Here you select the XSLT file you created earlier.

5. Click the first Browse button, select the *news.xsl* file, and then click the OK button in the Select XSLT File window.

The XSL Transformation window should now look like Figure PHP Tutorial 5-4. The XML file is automatically entered in the XML URI box. It will read either *feed.xml* (if you used the local file) or <http://www.nationalexasperator.com/feed.xml> (if you grabbed the XML file from the National Exasperator Web site.)

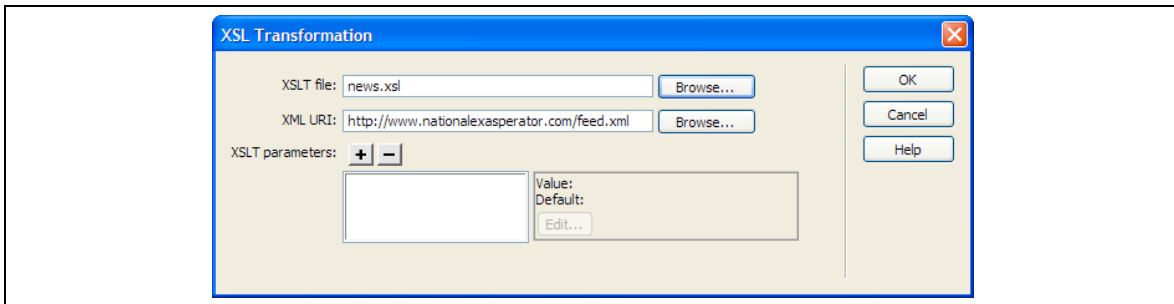


Figure PHP Tutorial 5-4. Use the XSL Transformation window to select the XSLT fragment you wish to add to the currently opened dynamic Web page.

6. Click OK to apply the Server Behavior to the page.

The document window should look like the top image in Figure PHP Tutorial 5-5.

7. Save the file and press F12 (Ctrl-F12) to see the results (Figure PHP Tutorial 5-5, bottom).

That's all there is to it. Expand on the steps in this tutorial using a real news feed from your favorite news site. (You can usually find these as links on the home page labeled "Feed," "RSS," or "XML"). If you can't find any feeds try this one from Wired.com: <http://www.wired.com/news/feeds/rss2/0,2610,,00.xml>.

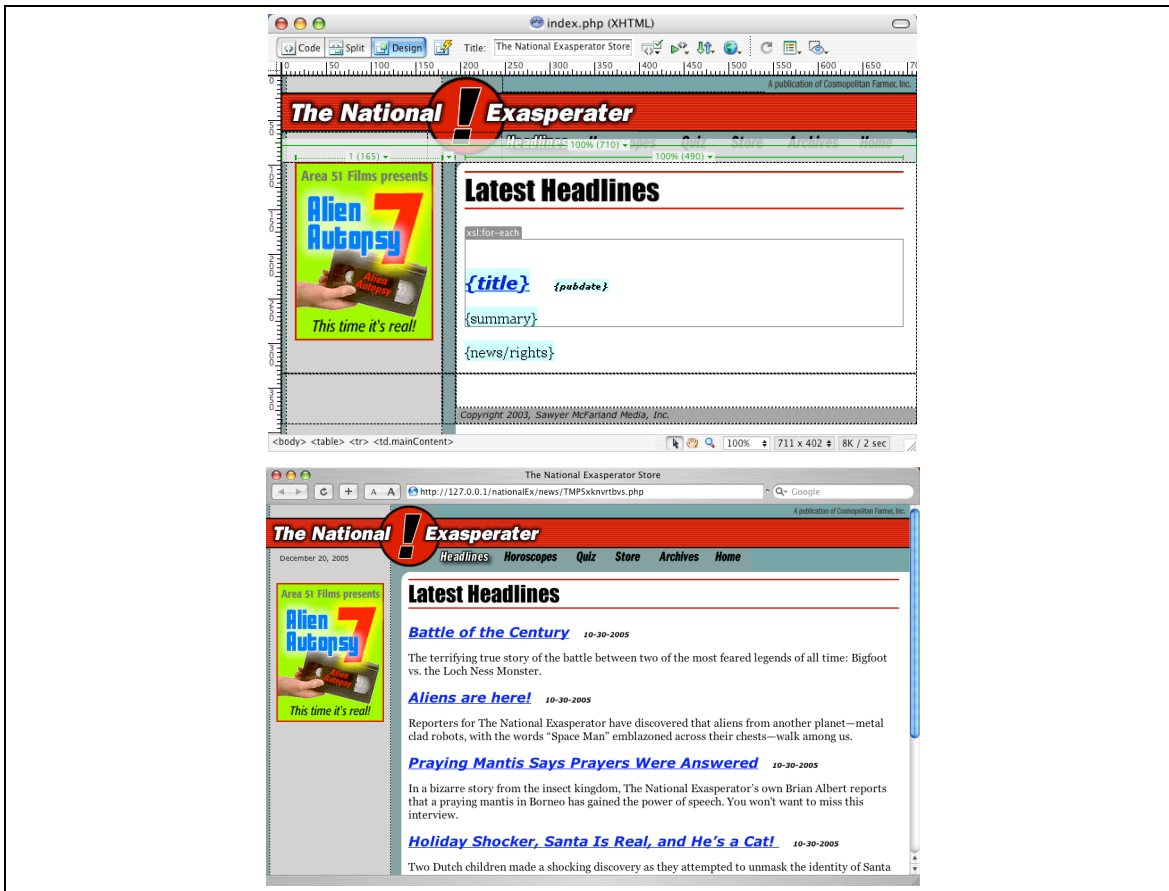


Figure PHP Tutorial 5-5. Although the XSLT fragment might not look like much in Dreamweaver (top), when viewed through a Web browser (bottom), it's magically transformed into a finished Web page, complete with headlines, summaries, links, and a copyright notice.

Copyright 2005, Sawyer McFarland Media, Inc. This material is supplied to readers of *Dreamweaver 8: The Missing Manual* as a helpful supplement to that book. If you don't own the book, what are you doing reading this? Please go out and buy the book: it's a good book, explains a lot more than this tutorial, and you'll feel better about yourself knowing you did the right thing. Thanks.