

XML Code documentation Lab

The door to your office flies wildly open and bangs against the wall, leaving an ugly mark. "Quick! hide under your desk!" your assistant warns. "The other programmers in the office love your business classes, but they don't know how to use them. Half of them have tar and the other half have feathers!"

"Not to worry," you calmly reassure, "I'll fix the problem. I'll simply edit my class files and put in some XML documentation comments."

"I don't think that'll do it. They don't just want commented code. Heck, some of them don't even have access to your source. They want a manual. And they want it now!"

"Again, don't fret.", you say. "Once I put those comments in, all I have to do is compile my code using the right compiler switch and it automatically generates a separate manual of glorious summaries, parameters, return values, remarks, examples, and exceptions."

"But, will it be readable by anyone?", he asks as he begins to calm.

"Anyone. Anywhere. Any time. It is XML. Who can't read XML?"

A smile begins forming on his face. "Will it be ... well-formed?"

"Yes, grasshopper, it will be well-formed," you say as you tousle his hair. "Crisis averted."

"Whew, that was a close one," He says as he visibly relaxes. "I'll spread the word. You get started on those comments."

Adding XML Documentation

1. Take a look at your business classes in your class library. For each class do these things:
 - Put a summary at the top of every class.
 - Go through each method in the class and give it a summary, return values, parameters, and an example. Add remarks where necessary.
2. Look at all of your remarks sections. You'll probably want to put paragraphs in `<para>` tags.
3. Go back through again and find all places where you've put code examples. Wrap those in a `<c>` or `<code>` tags.
4. Find all lists and put them in a `<list>` tag.

Intellisense

5. Go to any one of your forms in the UI part of your project. Try to instantiate one of your commented classes. Do you see your new descriptions in Intellisense? _____
6. Now try calling a method. You should see your documentation in Intellisense there also.
7. Make some changes to the comments and try again. Did Intellisense change? _____

Generating the documentation

8. Go to your project's properties. Under the Build tab, specify where you want your external XML documentation to be saved.
9. Save and build your project.
10. Find the XML documentation file. Open it and read through it.

Bonus! Not a trivial task, but if you have time, try to transform that file using an xslt transformation.

11. Ask your instructor for an xslt file and a css file that can be used to style the output.

12. Add a line like this after the XML declaration in your xml documentation file:

```
<?xml-stylesheet type="text/xsl" href="theXslFile.xsl" ?>
```

13. Run and test. Make any adjustments needed to tune it to your liking.