1 Extending Scaladoc to emit API docs in .NET format

Sandcastle[1] is "*an open source documentation generator from Microsoft that reads your assemblies (DLL or EXE files) and their XML Comments and automatically generates HTML documentation*[2]*.*"

Sandcastle carries out the very last phase of documentation generation, ensuring that the output meets the visual cues and file formats that developers expect of API docs on .NET. This division of labor bears similarities with the workflow of Scaladoc tools on JVM, where the final generation stage can be customized. That way, two goals can be achieved simultaneously:

- platform-friendliness (for example, the new .NET Lightweight style can be supported without additional effort);
- reuse of the Scaladoc front-end, which takes charge of parsing Scala sources and computing the document model of the resulting API documentation.

For comparison, the functionality of the Scaladoc front-end is usually embedded in .NET compilers (for example, as in C#[3] or F#[4]). In our case, we need not modify the Scala.NET compiler to emit documentation attributes in the output assemblies; and the required XML-comment file (with pointers back to source locations) is to be emitted by a new Scaladoc backend (a modular component) to be developed.

- Architecture of Scaladoc
  `http://wiki.scala-lang.org/display/SW/Scaladoc`,
  in particular the design description,
  `http://wiki.scala-lang.org/display/SW/Implementation+Details`
- Understanding Sandcastle,
  `http://en.wikipedia.org/wiki/Sandcastle_(software)`

1.1 A nice to have: Mono support

Regarding Mono, it's not clear whether the equivalent of Sandcastle (Monodoc) understands all the XML tags of Sandcastle. Therefore, it's not a requirement for this project to generate Monodoc-compatible XML files (it's still desirable, for example in case just a few tags can't be handled by Monodoc we can accomodate that). If our output works for Monodoc, fine. Quoting from [5]:

> *Mono generally uses Monodoc. As I understand it, this is a somewhat different format - but it can extract XML documentation from C# code as normal.*

- Monodoc, `http://www.mono-project.com/Monodoc`
- Monodoc's generating documentation:
  `http://www.mono-project.com/Generating_Documentation`

[1]`http://sandcastle.codeplex.com/`
[2]`http://broadcast.oreilly.com/2010/09/build-html-documentation-for-y.html`
[3]`http://msdn.microsoft.com/en-us/library/b2s063f7(v=vs.71).aspx`
[4]`http://msdn.microsoft.com/en-us/library/dd233217.aspx`
[5]`http://stackoverflow.com/questions/4451082/c-mono-api-documentation-tools`