This is NuDoc, a tool for generating documentation from C# code.

It all works by combining the information that can be pulled out of an assembly by reflection with the XML documentation files generated by the C# compiler.

Only the public API is exported: as slashdoc (for IntelliSense) and as simple html. (not html help!)

the benefits:

small, simple, fast.

easily integrated into an automated build. (msbuild task)

Project/Build properties: XML documentation file

it all starts here.

but then?

NDoc

Sandcastle

NDoc2, NDoc3

NDoc3: http://sourceforge.net/projects/ndoc3/

because Sandcastle is too slow. Agree!

jaha, ndoc3 kraschade. det var ju tråkigt. fast det hann åtminstone skriva en kanske bättre xml-fil först.

xml-formatet är dokumenterat här:

<http://msdn.microsoft.com/en-us/library/fsbx0t7x%28v=vs.100%29.aspx>

det som exporteras är bara själva kommentarerna; väldigt lite om typerna. skulle egentligen vilja dumpa ut hela det publika (och protected) gränssnittet – med reflection, förmodligen. finns det något verktyg för det? kanske kan plocka från ndoc?

Documentation comment XML files are known colloquially as <i>"SlashDoc"</i> files as they produced 23 /// by the Microsoft C# compiler when the /doc command-line option is specified. The format of these files is detailed in 24 /// the ECMA C# Specification (Appendix E).

NDoc source: most of the interesting stuff is in Core.Reflection.NDocXmlGenerator.

**TODO (backlog)**

* command line interface
* reporting of warnings (using Diagnostics.Trace?)
* error on missing summary
* ordna upp testklasserna; just nu finns det separata för C#, slashdoc, sample assembly. inte bra.
* plain old code review and refactoring.
* StyleCop
* ~~På msdn används “Equality”, “LessThanOrEqual” etc som namn på operatorerna. Se~~ [~~http://msdn.microsoft.com/en-us/library/643eft0t.aspx~~](http://msdn.microsoft.com/en-us/library/643eft0t.aspx)
* FxCop
* clean up source tree (e.g. move FxCop files)
* license
* readme

**Test run on TagLib-sharp.**

* ~~page title: assembly name~~
* ~~page header~~
* ~~maybe more space between types~~
* ~~type overview table needed (all types in assembly, with namespace)~~
* ~~generic class: don’t show the <T> in the header. E.g. ArrayIFDEntry<T> class – tried that, it didn’t work out.~~
* ~~expand cref’s. E.g. This class extends <see cref="Frame" />, implementing support for... (AttachedPictureFrame)~~
* ~~class File : File, System.IDisposable: this is pathological. Special case handling would just make things worse.~~
* ~~maybe put the class description headers in (invisible) tables, so that they line wrap like the tables do.~~
* ~~link the types in the type overview table to the type details (fragment url)~~
* ~~show type (display) names instead of slashdoc id’s in the summaries. need to back-translate somehow?~~

**Test run 2**

* the trivial methods (Equals, GetHashCode, etc) are present when overridden. They should be removed.
* “Provides data for the event.” – was “Provides data for the <see cref="E:TheClass.Finished"/> event.”; the simple but acceptable fix is to just remove the “E:”.
* a class which inherits from System.Collections.Generic.Dictionary<string, string> shows up with all the methods of the base class, but without any slashdoc. (It’s not in the slashdoc file so no coding error.)
  + this seems to apply to all inherited members. it could be worth trying a lookup of the base type in case the type itself doesn’t provide any slashdoc for it.

**Possible future extensions**

support the slashdoc <list> tag? see <http://msdn.microsoft.com/en-us/library/y3ww3c7e.aspx>

* some formatting for highlighting the member name could be useful
* show/hide members
* add links for urls?