

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

MVE - 2

Dokumentace k programu MMDoc

Vypracoval: **Petr Dvořák**
Datum: *15.prosince 2005*

1. Úvod

1.1. Účel tohoto dokumentu

Tento dokument obsahuje popis funkce a návrh implementace programu pro automatické generování dokumentace ke knihovným modulů *MVE2 projektu* s názvem *mmdoc*.

1.2. Obecné vlastnosti

Zadáním bylo vytvořit program, který automaticky generuje dokumentaci z XML souboru generovaného Microsoft Visual Studiem .NET a zkompileované knihovny *MVE2* modulů. Výsledná dokumentace by měla být uložena do jednoho souboru (*chm*, *mht* nebo *pdf*).

1.3. Používané zkratky a termíny

XML (eXtensible Markup Language) – Souhrn pravidel pro vytváření značkovacích jazyků typu HTML.

MHT (Mime HTML) – Tento formát slouží k zabalení celé HTML stránky včetně obrázků. Vychází z formátu elektronické pošty.

CHM – nápověda ve formátu HTML Help 1.x

HTML Help Compiler – Program od fy *Microsoft*, který vytváří *chm* nápovědu.

CSS (Cascading style sheets) – Kaskádní styl, používaný k definování vzhledu webových stránek.

XSLT (eXtensible Style Language Transformation) – Slouží pro převod mezi různými typy XML.

ADO (Microsoft ActiveX Data Objects) – Knihovny, které slouží k usnadnění přístupu k datům.

CDO (Collaboration Data Objects) – Tato knihovna v programu slouží k vytváření Mime HTML.

2. Implementace

2.1. Popis vstupu

Program používá informace z dokumentačních komentářů zdrojových kódů a metadat přikompilovaných ke knihovně. Prakticky to znamená, že jsou vstupem XML soubor a samotná knihovna s moduly. Oba musí mít stejný název (jméno XML souboru generovaného Visual Studiem se nastavuje v menu *Project->Properties->Configuration Properties->Build->XML Documentation File*).

2.2. Popis výstupu

Program generuje XHTML, CHM, DOC a MHT (Mime Html) dokumentaci. Lze zvolit i výstup do XML (spojené informace ze vstupního XML a knihovny), vstupních souborů pro HTML Help Compiler a XHTML v jednom souboru. Toto jsou ale spíše meziformáty používané

v programu (viz dále). Na druhou stranu jsou použitelné jinde (např. pro převod do HTML Help 2.x).

2.3. Architektura

System *mmdoc* byl původně vyvíjen pouze pro příkazový řádek. Ale možnosti *mmdoc* se stále rozšiřují a ovládání z příkazové řádky se stává složitější. Proto byl *mmdoc* projekt rozdělen na 3 projekty.

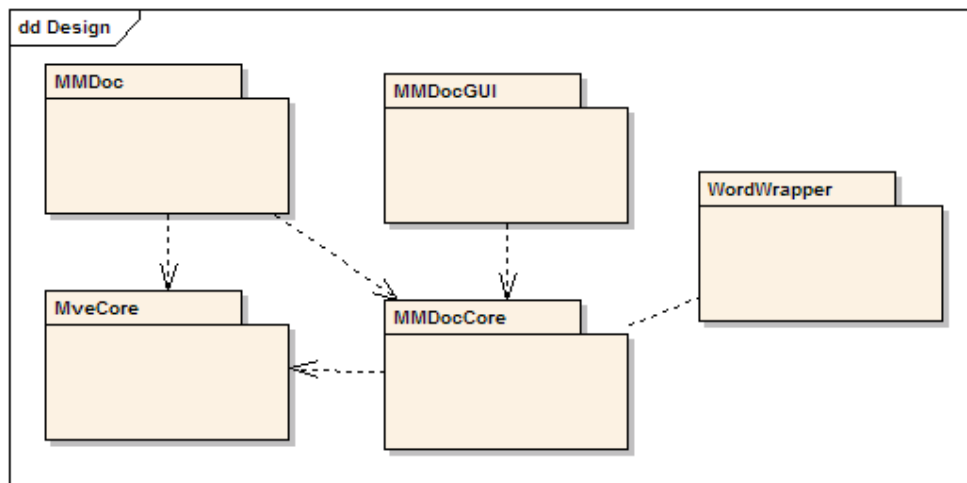
Samotný *mmdoc* nyní obsahuje jen příkazy pro ovládání z příkazové řádky. Vlastní jádro aplikace jako čtení a zpracovávání XML a knihoven, konverze atd. bylo přesunuto do *MMDocCore*. A nakonec třetím je grafické uživatelské rozhraní pro *mmdoc*.

Bylo třeba splnit několik podmínek:

1. Oddělené jádro *mmdoc* (*MMDocCore*) od programu pro ovládání z příkazového řádku (*MMDoc*) a z GUI (*MMDocGUI*).
2. Jednoduché rozhraní jádra při zachování možnosti dalšího rozšíření (např. při přidávání dalších parametrů pro generování).
3. Možnost uložení parametrů pro generování.
4. Jednotný způsob zobrazení výsledků případně chyb.

2.4. Design

Závislosti jednotlivých assembly:



Poměrně klíčová závislost je mezi *MMDocCore* a *MveCore* tj. jádrem *mmdoc* a *MVE2*. V *MveCore* jsou definovány *MVE2* atributy použité v dokumentaci dále třída *Module* a rozhraní *IDataObject*, pomocí nichž se hledají moduly a datové struktury, a nakonec *AbstractConfig* (pro práci s konfiguračními soubory).

Rozhraní jádra je tvořeno hlavně třídami `LibraryDoc`, která poskytuje většinu funkcionality, a `Options`, jež slouží k předání parametrů pro generování. `Options` tak může být jednoduše rozšířena o další parametry, aniž by se změnilo rozhraní `LibraryDoc`.

Obdoba `Options` v GUI je `OptionsGUI`. V podstatě je téměř stejná. Chybí jí serializační metody (viz dále) a atributy nutné pro GUI (komponentu `PropertyGrid`).

`MMDocGUI` má více možností než `mmdoc`. Je zde lepší kontrola vstupu (tj. nedovolí zadat knihovnu bez XML) a možnost uložit a načíst parametry pro kompilaci dokumentace. K tomu se používá XML serializace.

Hlášení chyb a výsledků zajišťuje jádro výpisem na konzoly. V GUI je tento výstup přesměrován do textového pole (`RichTextBox`). To se provádí oddělením od `StringWriter` a přetížením metod `Write` a `WriteLine`. Taková třída lze pak použít pro přesměrování příkazem `Console.SetOut`.

`WordWrapper` je assembly, která umožňuje přístup ke komponentě `Wordu`. Je oddělena od jádra, aby se odstranila závislost jádra na instalaci `Wordu`.

2.5. Analýza jádra

Během analýzy jsem dospěl k několika kritériím, které bylo třeba splnit:

1. Pro jednu knihovnu jsou dva datové vstupy (tj. xml a dll knihovna)
2. Vstupem může více knihoven
3. Je vhodné propojit knihovny odkazy
4. Snadné úpravy (na začátku se nevědělo co všechno bude výsledná dokumentace obsahovat), zachování přehlednosti
5. Rozšíření o další výstupní formáty

Proto bylo odděleno zpracování informací a jejich prezentace tj. zpracování do XML a jejich následný převod pomocí XSLT a CSS. Chceme-li vygenerovat další XHTML, pak jen stačí napsat XSLT šablonu (bez zásahu do zdrojových kódů).

Kvůli bodu propojení knihoven odkazy bylo třeba všechny informace z XML a DLL načíst, následně projít a porovnat (propojení knihoven odkazy, kontrola zdrojů) a teprve potom vytvořit XML, které se může dále nezávisle zpracovávat.

Tento způsob má i nevýhody:

- pro větší XML soubory může být o něco pomalejší
- napsání XSLT šablon není triviální.

2.6. Algoritmus řešení

Program si v prvním kroku vstupní informace z knihovny a XML uloží do jedné XML datové struktury.

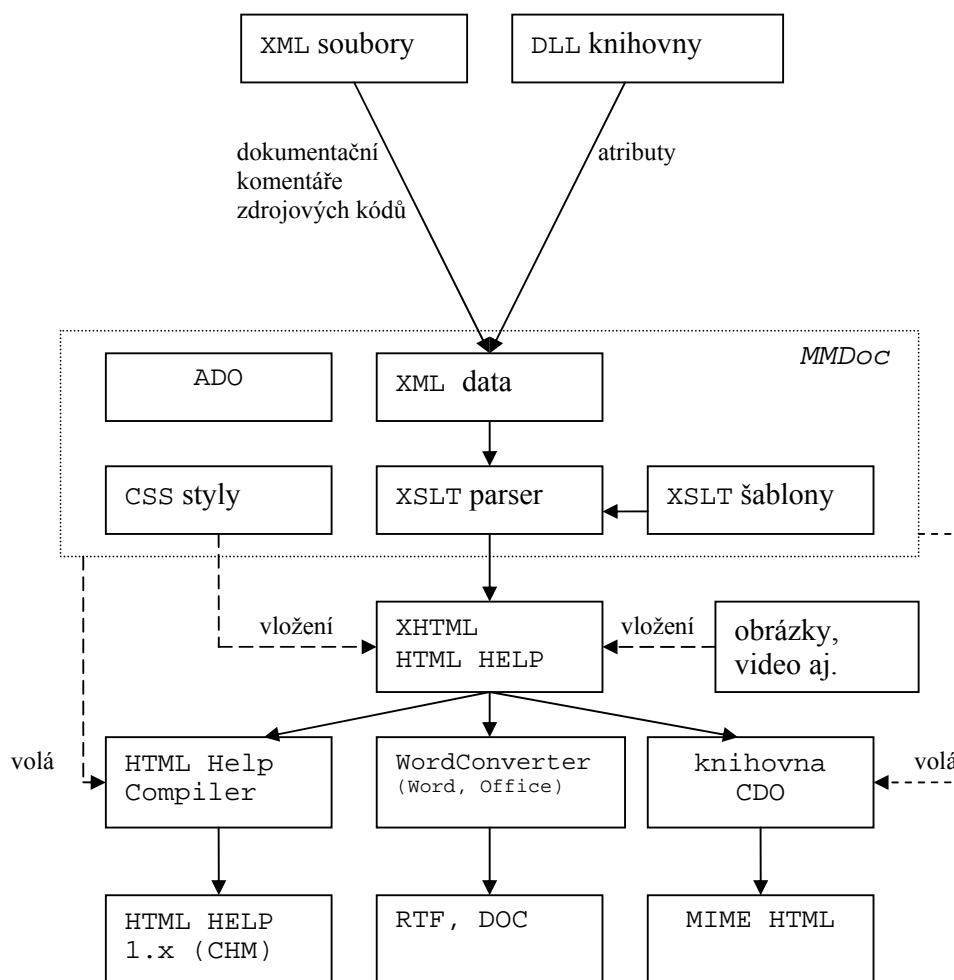
V dalším kroku se tato XML data předají XSLT parseru společně s XSLT šablonami a vygeneruje se několik XHTML souborů podle počtu modulů a datových struktur. Eventuelně se

vygenerují ještě projektové soubory pro HTML Help Compiler. Ten je v posledním kroku spuštěn a vytvoří jeden chm soubor. Podobně lze zvolit vytvoření MHT za pomoci ADO a CDO a DOC (RTF) použitím wrapperu pro *MS Word* pojmenovaný *WordConverter*.

Struktura výsledných souborů je dána XSLT šablonami a vzhled CSS styly.

V dokumentačních komentářích mohou být použity i HTML tagy. To umožní mimo jiné vložit do dokumentace i obrázky pomocí tagu `img ()`. Tyto obrázky budou součástí výsledné dokumentace.

2.7. Diagram činnosti



2.8. Popis tříd jádra

`LibraryDoc`

Souhrn informací a metod pro vytvoření dokumentace všech zadaných knihoven. Volá transformace XML XSLT šablonami (asi 13 šablon).

`Option`

Obsahuje parametry předané z uživatelského rozhraní jádru.

`XmlResourceResolver`

Jde o `XmlResolver`, který načítá inkludované XSLT šablony jako resource (standardní způsob je jako soubor).

`ChmCompiler`

Spouští *HTML Help Compiler*.

`Library`

Souhrn informací a metod pro jednu knihovnu.

`XmlBaseEntity`

Abstraktní třída se společnými metodami modulů a datových struktur pro vytváření XML.

`XmlDataObject`

Třída pro vytvoření XML datové struktury.

`XmlModule`

Třída pro vytvoření XML modulu.

`CustomMembers`

Třída zpracuje informace o metodě pro snadné vložení do XML.

`CustomPropertyInfo`

Třída zpracuje informace o property pro snadné vložení do XML.

`Speed`

Měří čas běhu ohraničené části kódu. Primárně měří čas běhu. Lze použít i pro ladící účely.

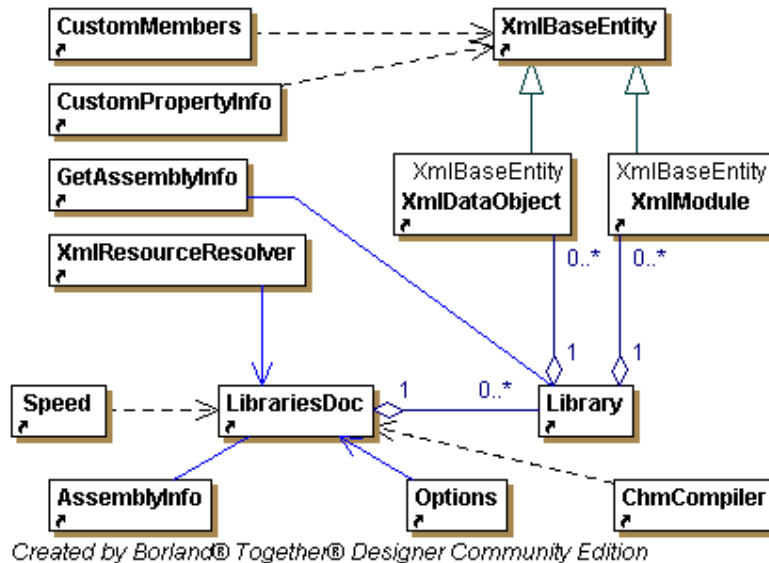
`AssemblyInfo`

Informace o `mmdocu` reprezentované atributy.

`GetAssemblyInfo`

Obsahuje metody pro získání atributů knihoven.

2.9. UML jádra



3. Uživatelská dokumentace

3.1. Podmínky pro správnou činnost

Aby program vytvořil kvalitní dokumentaci je třeba při vytváření knihovny dodržovat konvence pro komentování popsané v dokumentu *Komentování MVE2 knihovny.doc*.

Dále je třeba se držet pravidel pro tvorbu modulů a datových struktur. Zejména to, že moduly musí být potomky (přímými nebo nepřímými) třídy `Zcu.Mve.Core.Module` a datové struktury musí implementovat rozhraní `Zcu.Mve.Core.IDataObject`.

3.2. Popis programu

Program vytvoří dokumentaci k *MVE2* knihovně.

Vstupem je XML soubor a samotná knihovna s moduly. Oba musí mít stejný název (jméno XML souboru generovaného Visual Studiem se nastavuje v menu *Project->Properties->Configuration Properties->Build->XML Documentation File*).

Program generuje XHTML, CHM, DOC a MHT (Mime Html) dokumentaci. Lze zvolit i výstup do XML (spojené informace ze vstupního XML a knihovny), vstupních souborů pro HTML Help Compiler a XHTML v jednom souboru.

Existují dvě verze pro:

- příkazový řádek
- grafické uživatelské rozhraní

3.3. *MMDoc* v příkazovém řádku

3.3.1 Spuštění

Příklad spuštění kompilace:



```
C:\WINDOWS\system32\cmd.exe
D:\Projekty\mve\svn\Mve2-src-2005\MMDoc\bin\Debug>mmdoc.exe -l Examples -dir ./img -name lib-doc
Start of documentation generating.
Start of loading Examples types.
End of loading Examples types.
Start of joining information about libraries into one XML file.
Start of converting Examples types into XML.
End converting Examples types.
End of joining information about librariers.
Saving XML.
Start of moving files from D:\Projekty\mve\svn\Mve2-src-2005\MMDoc\bin\Debug\img
into D:\Projekty\mve\svn\Mve2-src-2005\MMDoc\bin\Debug\lib-doc\.
End of moving files.
Start of HTML files generating.
  index-Examples.html
  index.html
  Examples.TetraMeshSlicer.html
  Examples.TetraMeshSlicer-dialog.html
  Examples.TetraMeshSlicer-members.html
  Examples.PointToTetrahedronDistanceSampler.html
  Examples.PointToTetrahedronDistanceSampler-dialog.html
  Examples.PointToTetrahedronDistanceSampler-members.html
  Examples.ObjLoader.html
  Examples.ObjLoader-dialog.html
  Examples.ObjLoader-members.html
  Examples.NormalComputer.html
  Examples.NormalComputer-dialog.html
  Examples.NormalComputer-members.html
  Examples.VertexEdgeDetector.html
  Examples.VertexEdgeDetector-dialog.html
  Examples.VertexEdgeDetector-members.html
  Examples.PointToTetrahedronDistanceEvaluator.html
  Examples.PointToTetrahedronDistanceEvaluator-dialog.html
  Examples.PointToTetrahedronDistanceEvaluator-members.html
  Examples.TetraMeshComposer.html
  Examples.TetraMeshComposer-dialog.html
  Examples.TetraMeshComposer-members.html
  Examples.Sumator.html
  Examples.Sumator-dialog.html
  Examples.Sumator-members.html
  Examples.PromennePorty.html
  Examples.PromennePorty-dialog.html
  Examples.PromennePorty-members.html
  Examples.FrameExtractor.html
  Examples.FrameExtractor-dialog.html
  Examples.FrameExtractor-members.html
  Examples.Sinus.html
  Examples.Sinus-dialog.html
  Examples.Sinus-members.html
  Examples.GenerateGraph.html
  Examples.GenerateGraph-dialog.html
  Examples.GenerateGraph-members.html
  Examples.NumberSource.html
  Examples.NumberSource-dialog.html
  Examples.NumberSource-members.html
  Examples.Convolution.html
  Examples.Convolution-dialog.html
  Examples.Convolution-members.html
  Examples.AllEventSumator.html
  Examples.AllEventSumator-dialog.html
  Examples.AllEventSumator-members.html
  Examples.ScalarNumber.html
  Examples.ScalarNumber-members.html
End of generating.

Documentation was successfully generated.
Time: 2.42 s
```

`mmdoc` pro příkazový řádek je určen hlavně pro dávkové zpravování. Spouští se příkazem:


```
mmdoc.exe -l modul
```

Např.:

```
mmdoc.exe -l MveCore
```

- vytvoří dokumentaci k *MVECore.dll* v adresáři *MVECore*

```
mmdoc.exe -l MveCore -l Visualization
```

- vytvoří dokumentaci k *MveCore.dll* a *Visualization.dll* v adresáři *MveCore*
- je vhodné (ale není to podmínkou), aby obě knihovny byly ve stejném adresáři

```
mmdoc.exe -l MveCore -name Jadro
```

- vytvoří dokumentaci k *MveCore.dll* v adresáři *Jadro*

```
mmdoc.exe -l MveCore -title „Title of the documentation“
```

- umožní definovat vlastní nadpis dokumentace

```
mmdoc.exe -l MveCore -w
```

- pokud někde chybí komentář nebo ztrát vypíše chybové hlášení

```
mmdoc.exe -l MveCore -noprotected
```

- do výsledné dokumentace nebudou zahrnuty *protected* metody, *property* atd.

3.3.2. Určení výstupního formátu

```
mmdoc.exe -l MveCore -html
```

- vytvoří *html* dokumentaci k *MVECore.dll* v adresáři *MVECore*

```
mmdoc.exe -l MveCore -chm
```

- vytvoří *chm* nápovědu k *MveCore.dll*
- aby tato možnost fungovala, musí být na počítači nainstalován *Microsoft Html Help Workshop* nebo se musí nacházet ve stejném adresáři s *mmdoc.exe* i *Html Help Compiler* (ten je součástí *HTML Help Workshopu*)

```
mmdoc.exe -l MveCore -mht
```

- vytvoří dokumentaci ve formátu *mht*

```
mmdoc.exe -l MveCore -hhc
```

- vytvoří dokumentaci a projektové soubory pro *Html Help Compiler* k *MVECore.dll*

```
mmdoc.exe -l MveCore -xml
```

- vytvoří *xml* soubor s informacemi z *MveCore.xml* a atributů z *MveCore.dll*

```
mmdoc.exe -l MveCore -lhtml
```

- vytvoří *html* dokumentaci s jedním souborem *html*

```
mmdoc.exe -l MveCore -doc
```

- vytvoří *doc* dokumentaci pro otevření v *Microsoft Wordu*.

```
mmdoc.exe -l MveCore -rtf
```

- vytvoří `rtf` dokumentaci pro otevření v *Microsoft Wordu*.

Tyto přepínače se mohou vynechat. Jako výchozí se vezme poslední použitý.

3.3.3. Vložení obrázků

Autor knihovny může v dokumentačních komentářích použít tag ``, kde určí relativní cestu k obrázku. Přepínačem `-dir` se pak určí k jakému adresáři se tyto relativní cesty vztahují. Lze samozřejmě použít i absolutní cestu.

```
mmdoc.exe -l MveCore -dir "D:\\img"
```

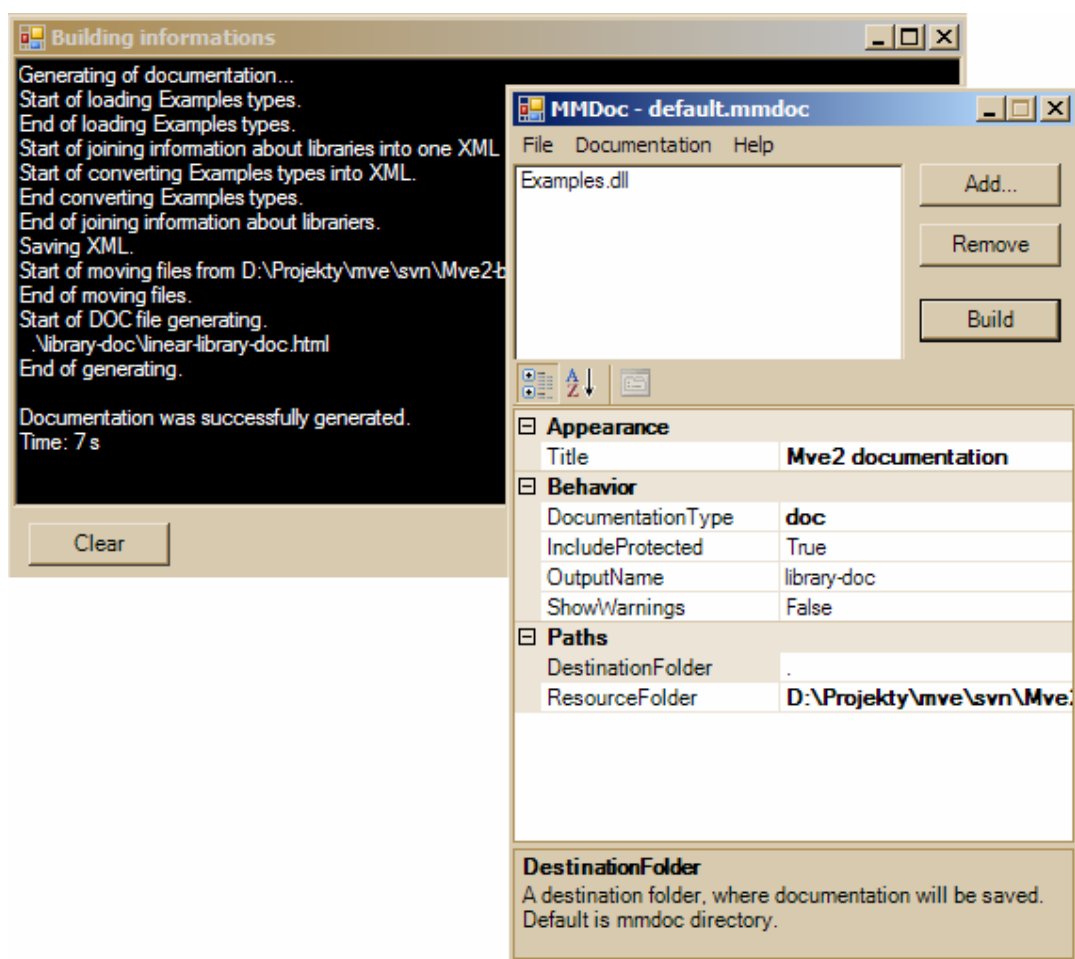
- vytvoří dokumentaci k `MveCore.dll` v adresáři `MveCore` a nakopíruje do něj obsah `D:\img`

Příklad: V dokumentačních komentářích k `MveCore` jsou uvedeny tagy `` a ``. Aby byl výše uvedený příkaz korektně zpracován musí být v adresáři `D:\img` soubor `delay.jpg` a adresář `Examples` se souborem `sumator.jpg`.

Pozn.: Jinou možností je přepínač `-dir` vynechat a obrázky prostě jen nakopírovat do výstupního adresáře.

3.4. Grafické uživatelské rozhraní pro *MMDoc*

Grafické uživatelské rozhraní zjednodušuje použití *mmdoc* a zadávání parametrů kompilace.



V menu *File* lze otevřít, uložit a vytvořit nový projekt. Projekt obsahuje informace nutné pro kompilaci jako je typ dokumentace, cesta ke vstupním souborům atd. Tyto informace se nastavují v dolní části hlavního okna (v *PropertyGridu* známého např. z *Visual Studio 2003*).

Poznámka: Pro vytváření *chm* dokumentace je třeba nainstalovat *HTML Help Workshop*.

Seznam knihoven, pro něž se dokumentace vytváří, se upravuje v horní části hlavního okna tlačítky *Add...* a *Remove*.

Vlastní kompilace spustí výberem *Documentation->Build* v menu nebo klávesou *F5*. Objeví se okno s informacemi o probíhající kompilaci.

Pokud se vyskytnou nějaké problémy, navštivte domovkou stránku *MVE2* projektu volbou menu *Help->MVE2 Online*.

4. Závěr

Stále přibývá spousta vychytávek. Od třetí verze to je dobře použitelná kompaktní verze (např. styly i šablony jsou přikompilovány v `mmdoc.exe`). Další verze rozšiřují generovanou dokumentaci o další informace (např. seznam konstruktorů, metod, properte, členských proměnných, rozhraní, zrychlení generování atd.).

5. Odkazy

Html Help Workshop: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/htmlhelp/html/hwMicrosoftHTMLHelpDownloads.asp>

6. Přílohy

6.1. *Příklady výsledné dokumentace*

Popis knihovny ve výsledné dokumentaci:



MVE2 Module Documentation

Examples

Title: *Modular Virtual Environment 2 - Example*

Description: *Example of MVE2 library.*

Company: *University of West Bohemia in Pilsen, Czech Republic*

Copyright: © 2004 - 2005 *University of West Bohemia, Czech Republic*

Version: *1.0.1936.19787*

Runtime version: *1.1.4322*

Modules

Examples.AllEventSumator	Sample module. It sums two input numbers. Reaction to all events is implemented.
Examples.Convolution	Module performs a convolution of a image.
Examples.GenerateGraph	Module creates 100 samples of connected function in $\langle 0, 10 \rangle$ interval.
Examples.PromennePorty	Sample module. Example of varying ports.
Examples.NumberSource	Sample module. Source of one random scalar number.
Examples.Sinus	Module calculates sinus of given number.
Examples.Sumator	Sample module. It sums two input numbers.

Data structures

Examples.ScalarNumber	One scalar number (double)
---------------------------------------	----------------------------

Popis modulu ve výsledné dokumentaci:



MVE2 Module Documentation

Examples.ScalarNumber

[Index](#) | [Overview](#) | [Members](#)

Sample data class. It represents one scalar number (accuracy as double type in .NET).
Can be shared by modules.

Inheritance:

```
System.Object
  Zcu.Mve.Core.DataObject
    Examples.ScalarNumber
```

Implements:

```
Zcu.Mve.Core.IDataObject
```

Created by MMDoc 20.4.2005 12:00:08

Nápověda ke konfiguračnímu oknu modulu:



MVE2 Module Documentation

Examples.NumberSource



[Index](#) | [Overview](#) | [Settings](#) | [Events](#)

You can choose from two modes of function of module:

1. Generating of random number:



2. Generating of defined number (2.0 in this case):



Created by MMDoc 12/12/2005 8:59:58 PM