

AWPBTools Help Compiler

Version 0.1

Freeware

Dieses Programm verwendet Code der auf dem expat XML Parser basiert. (<http://expat.sourceforge.net>)

(c) 2014 by Ronny Krüger

Beschreibung

Dieses Programm dient dazu, um eine Dokumentationsdatei im XML Format in HTML-Dateien zu übersetzen, die online oder lokal im Browser betrachtet werden können, oder mit einem entsprechenden Compiler in CHM Dateien überführt werden können.

Projektdateien (Projekt, Index, TOC) für den MS Help Workshop werden automatisch erzeugt.

Das Programm selber läuft unter Windows und Linux (sowohl 32 als auch 64 Bit).

Benutzung

Das Programm kann aus dem Explorer heraus gestartet oder in der Kommandozeile aufgerufen werden.

Beim Aufruf vom Explorer wird ein Dateidialog zur Auswahl der Dokumentations-Datei angezeigt sowie ein Ordnerauswahldialog um den Zielordner zu wählen und ggf zu erstellen.

Beim Aufruf aus der Kommandozeile wird der Dateiname direkt mit übergeben, die HTML-Dateien werden im aktuellen Pfad angelegt.

Aufbau der Dokumentationsdatei

Es handelt sich hierbei um eine standardkonforme XML-Datei im UTF8-Format.

Der strukturelle Aufbau ist dem nachfolgenden Muster zu entnehmen.

Angaben wie z.B.

```
<element></element>
...
<element></element>
```

bedeuten, das es in dieser Ebene unendlich viele element-Tags geben kann.

Die Reihenfolge der Tags bzw Attribute ist nicht wichtig, jedoch müssen Tagnamen und Attributnamen klein geschrieben werden.

Grundstruktur

```
<doc>
  <author></author>
  <license></license>
  <copyright></copyright>
  <description></description>
  <library>
    <keywords></keywords>
    <description></description>
    <structure>
      <keywords></keywords>
      <description></description>
      <element></element>
      ...
      <element></element>
    </structure>
    ...
    <structure>...</structure>
    <constant>
      <keywords></keywords>
      <description></description>
    </constant>
```

```

...
<constant> ... </constant>
<command>
    <keywords></keywords>
    <description></description>
    <parameter>
        <arg></arg>
        ...
        <arg></arg>
    </parameter>
    <result></result>
    <references>
        <reflibrary />
        ...
        <reflibrary />
        <refclass />
        ...
        <refclass />
        <refcommand />
        ...
        <refcommand />
        <refconstant />
        ...
        <refconstant />
        <refstruct />
        ...
        <refstruct />
    </references>
    <os></os>
    ...
    <os></os>
    <example></example>
    ...
    <example></example>
</command>
...
<command> ... </command>
<class>
    <keywords></keywords>
    <description></description>
    <command> ... </command>
    ...
    <command> ... </command>
</class>
...
<class> ... </class>
</library>
...
<library> ... </library>
</library>
</doc>

```

Attribute der jeweiligen Tags:

```

<arg index="" name="" type="" default="">Text</arg>
    index - Position des Parameters (das erste hat die Nummer 1)
    name  - Name der Variable
    type  - Typ der Variable (Optional) (ohne .)
    default - Vorgabewert falls es eine optionale Variable ist (Optional)
    Text  - Beschreibung

```

```

<author>Text</author>
    Text - Name des Autors, Adresse etc

```

```

<class name=""></class>
    name - Bezeichnung der Klasse

```

```

<command name=""></command>
    name - Bezeichnung der Prozedur/Funktion (ohne Klammern)

```

```

<constant name=""></constant>
    name    - Bezeichnung (ohne #)

<copyright>Text</copyright>
    Text    - Copyright-Text

<description>Text</description>
    Text    - Beschreibung

<doc lang="" title=""></doc>
    lang    - Sprache (z.B. 0x407 für Deutsch) (optional)
    title   - Titel des Projektes

<element offset="" name="" type="">Text</element>
    offset  - Offset im Speicher
    name    - Bezeichnung des Elements
    type    - Typ des Elements (ohne .) (optional)
    Text    - Beschreibung

<example name="">Text</example>
    Text    - Quellcode, wird 1:1 übernommen, inkl. führende Leerzeichen

<keywords>Liste</keywords>
    Liste   - mit ";" getrennte Schlüsselwörter für den Index

<library name=""></library>
    name    - Bezeichnung der Bibliothek

<license>Text</license>
    Text    - Lizenztext

<os>Text</os>
    Text    - Bezeichnung des Betriebssystems, das von dem Befehl unterstützt wird. Wird bei einem Command
               kein OS angegeben, ist die Vorgabe "alle".

<parameter></parameter>

<refclass name="" lib="" />
    name    - Bezeichnung der Klasse
    lib     - Bibliothek, die die Klasse enthält

<refcommand name="" class="" lib="" />
    name    - Bezeichnung der Funktion/Prozedur
    class   - Klasse, muss angegeben werden, wenn es sich um eine Memberfunktion einer Klasse handelt
    lib     - Bibliothek, die die Funktion (ggf. inkl. Klasse) enthält

<refconstant name="" lib="" />
    name    - Bezeichnung der Konstante
    lib     - Bibliothek, die die Konstante enthält

<references></references>

<reflibrary name="" />
    name    - Bezeichnung der Bibliothek

<refstructure name="" lib="" />
    name    - Bezeichnung der Struktur
    lib     - Bibliothek, die die Struktur enthält

<result name="" type="">Text</result>
    name    - Bezeichnung
    type    - Typ

```

Text - Beschreibung

```
<structure name=""></structure>  
name - Bezeichnung
```

Jede Beschreibung die im Body eines Tags erfolgt, darf keine HTML-Tags enthalten. Wird eine solche benötigt, muss der Text zusätzlich in CDATA-Tags eingeschlossen werden, z.B.:

```
<description>  
  <![CDATA[  
    Text mit HTML-Tags  
  ]]>  
</description>
```

Mehrere Leerzeichen werden analog zu HTML zu einem zusammengefasst. Zeilenumbrüche werden ignoriert. Um einen Zeilenumbruch oder Absatz zu erzeugen siehe unter Formatierungsangaben.

Formatierung von Beschreibungen

Beschreibungen können begrenzt formatiert werden, ohne HTML zu benutzen. Dies funktioniert sowohl innerhalb als auch ausserhalb von CDATA-Tags.

Die Zeichen <, > und @ sowie explizite Zeilenumbrüche bzw Leerzeichen müssen maskiert werden, da sie sonst vom Parser missverstanden werden.

Nach bzw vor den Codes ist kein Leerzeichen erforderlich, es sei denn, dort soll eines stehen.

Folgende Möglichkeiten existieren:

Referenzen (anklickbare Links):

```
@refconstant(Konstantenname, Bibliothek)  
@refclass(Klassenname, Bibliothek)  
@refcommand(Prozedurname, [Klasse], Bibliothek)  
@refstruct(Structurname, Bibliothek)  
@reflibrary(Bibliothekensname)
```

Bis auf Klasse bei @refcommand sind alle Parameter Pflichtangaben. Es dürfen weder Klammern bei @refcommand noch # bei @refconstant angegeben werden.

Optische Auszeichnungen:

Konstante hervorheben (ohne Verknüpfung), das # darf nicht angegeben werden

```
@const (Konstantenname)
```

Funktion/Prozedur hervorheben (ohne Verknüpfung), Klammern dürfen nicht angegeben werden

```
@func (Function)
```

Quellcode farblich und mit Mono-Schriftart anzeigen

```
@code ... @endcode
```

Text fett darstellen

```
@bold ... @endbold
```

Text kursiv darstellen

@italic ... @enditalic

einen Absatz erzeugen

@paragraph ... @endparagraph

weitere Formatierungsangaben:

Zeilenumbruch

@linebreak

@-Zeichen ausgeben

@@

< ausgeben

@lt

> ausgeben

@gt

Leerzeichen erzwingen

@spc

Beispiel

```
<doc lang="0x407" title="Tools">
  <author>
    Name des Autors
  </author>
  <license>
    Lizenz des Sammlung
  </license>
  <copyright>
    (c) by mir
  </copyright>
  <description>
    Beschreibung des Gesamtpaketes
  </description>
  <library name="Bibliothek1">
    <keywords>
      Bibliothek; Code; Scanner; Parser
    </keywords>
    <structure name="Struktur1">
      <description>
        <![CDATA[
          Beschreibung der <b>Struktur</b><br><br>
          In diesem Fall mit HTML-formatierten Code.
          Siehe auch: @reflibrary(Bibliothek2)
        ]]>
      </description>
      <element offset="0" name="index" type="i">Index</element>
      <element offset="4" name="*name">Zeiger auf Name</element>
      <element offset="8" name="size" type="l">Größe des Inputs</element>
    </structure>
    <constant name="Konstante1">
      <description>Eine Konstante.</description>
      <keywords>
        Const; Parser
      </keywords>
    </constant>
  </library>
</doc>
```

```

        </keywords>
</constant>
<constant name="Konstante2">
    <description>Noch eine Konstante.</description>
    <keywords>
        Scanner; Parser
    </keywords>
</constant>
<command name="Prozedur1">
    <description>
        Diese Prozedur macht eigentlich das selbe wie diese hier:
        @refcommand(Prozedur2,,Bibliothek1) @linebreak
        die Prozedur @refcommand(Prozedur1, Klasse1, Bibliothek1) macht ebenfalls
        das selbe. Dazu siehe auch die Klasse @refclass(Klasse1,
        Bibliothek1).@linebreak
        In @reflibrary(Bibliothek2) werden @italicähnliche@enditalic Funktionen
        enthalten sein.
        @paragraph
        Ein @bold extra Paragraph @endbold.
        @endparagraph
        @paragraph
        Und noch einer; @@ wird in eMails benötigt.
        @lt und @gt werden in HTML benötigt.
        @endparagraph
    </description>
    <parameter>
        <arg index="1" name="index" type="i">Index</arg>
        <arg index="2" name="text" type="s">der zu bearbeitende Text</arg>
        <arg index="3" name="*target" default="#Null">
            @boldZeiger@endbold auf das Ziel, optional
        </arg>
    </parameter>
    <result name="result" type="i">
        @const(True) bei Erfolg, sonst @const(False)
    </result>
    <references>
        <reflibrary name="Bibliothek2" />
        <refclass name="Klasse1" lib="Bibliothek1" />
        <refcommand name="Prozedur1" class="Klasse1" lib="Bibliothek1"/>
        <refconstant name="Konstante1" lib="Bibliothek1" />
        <refstructure name="Struktur1" lib="Bibliothek1" />
    </references>
    <os>Windows</os>
    <os>Linux</os>
    <example name="test1.pb">
; Muss links anfangen, da 1:1 übernommen
For a = 1 to 20
    b + 1
Next
    </example>
</command>
<command name="Prozedur2">
    <description>
        Nur diese Beschreibung.
    </description>
</command>
<class name="Klasse1">
    <command name="Prozedur1">
        <description>
            Dies ist @func(Prozedur1) in @code Klasse1@endcode.@linebreak
            Nützlich ist ein Verweis auf @refstruct(Struktur1, Bibliothek1)
        </description>
        <parameter>
            <arg index="1" name="index" type="i">Index</arg>
            <arg index="2" name="text" type="s" default='"Standardwert"'>
                der zu bearbeitende Text
            </arg>
            <arg index="3" name="*target" default="#Null">
                @boldZeiger@endbold auf das Ziel, optional
            </arg>
        </parameter>
        <result name="*result">Zeiger auf das Ergebnis</result>
        <os>Windows</os>
        <keywords>

```

```
Klassel; Prozedurl  
    </keywords>  
</command>  
<description>  
    Dies ist eine Klasse mit einer Prozedur.@linebreak  
    Konstanten und Strukturen können nicht in einer Klasse definiert  
    werden.@linebreak  
    Funktionen sind identisch in der Definition wie außerhalb.  
</description>  
</class>  
<description>  
    Dies hier ist Bibliothek1, die Funktionen, Klassen, Strukturen und Konstanten  
    enthält.@linebreak  
    @italicDas alles ist hier drunter aufgelistet.@enditalic  
</description>  
</library>  
<library name="Bibliothek2">  
    <description>  
        Eine zweite      Bibliothek des Toolsets, diese enthält aber bis auf die  
        Beschreibung noch nichts.@linebreak  
        Diese Zeile enthält wirklich @spc@spc@spc@spc@spc@spc@spc@spc mehrere  
        Leerzeichen.  
    </description>  
    <keywords>Bibo2</keywords>  
</library>  
</doc>
```