

pisa

Simple PDF generator

(C)opyright by spirito GmbH, 2005-2007

Einführung

PDF Markup Language (**PML**) ist ein SGML-Dialekt der HTML in vielerlei Hinsicht ähnelt und ist sogar in Grundzügen kompatibel damit. PML baut auf dem Toolkit für PDF-Generierung der Firma ReportLab (<http://www.reportlab.com/>) auf. Weitere interessante technische Details finden sich in der Dokumentation zu deren Toolkit.

Befehle, die nicht erkannt werden, werden vom System einfach ignoriert. Es sollte also möglich sein - vorausgesetzt es gibt keine syntaktischen Fehler - auch komplexe HTML Seiten einfach durch PML zu jagen.

Kommandozeilentool

Für die ersten Tests bietet sich die Verwendung des Kommandozeilentools an. Unter Windows wechseln sie in das Kommandofenster und dann in das Verzeichnis in dem sich die Datei "pisa.exe" befindet, oder sie binden das Tool in den Suchpfad ein. Nun erzeugen sie ein Datei "test.pml" in der sie ein einfaches Demo in PML programmieren. Es folgt der eigentliche Übersetzungsvorgang, z. B. so :

```
pisa test.pml test.pdf
```

Als Ergebnis erhalten sie die Datei "test.pdf", die sich dann im AcrobatReader ansehen können. Geben Sie "pisa -h" ein um alle weiteren verfügbaren Optionen zu sehen.

Standalone oder als Python Modul

Installation

ReportlabLib: Anschließend muß noch die aktuellste reportlablib ab der Version 1.20 installiert werden. Dazu laden sie dieses Programmpaket von der Adresse <http://reportlab.org/ftp/current.tgz> herunter und folgen den Installationsanweisungen.

PIL: Zum reibungslosen Betrieb, besonders im Zusammenhang mit dem Einbinden von Bildern in ein PDF empfiehlt sich die Installation der PIL Library für Python (<http://www.pythonware.com/products/pil/>).

pyPDF: Des weiteren wird für die Verwendung von Wasserzeichen die Python Library pyPDF benötigt (<http://pybrary.net/pyPdf/>).

Benutzung

Schreiben sie ein kleines Python Programm, das wie folgt aufgebaut ist:

```
import spirito.pisa
pdf = spirito.pisa.CreatePDF("Hallo <b>Welt</b>", open("test.pdf", "wb"))
```

Nachdem sie das Programm gespeichert und ausgeführt haben wird eine PDF Datei mit dem Namen test.pdf erzeugt, in der "Hallo **Welt**" steht.

Beispiele

Das obligatorische Hello-World-Beispiel:

```
Hello World
```

Etwas komplexer:

```
<document>

<template name="second">
<frame box="1cm 1cm 10cm 10cm">
</template>
```

```

<style font="Times-Roman">

Hallo:
<ol>
<li> erstens
<li> zweitens
</ol>

<nexttemplate name="second">
<nextpage>

Oben links
</document>

```

Koordinatensystem

Üblicherweise liegt das Koordinatensystem bei PDF-Dateien in der Ecke unten links. Für PML wurde der Ursprung in die Ecke oben links verlegt, um das visuelle Arbeiten wie bei HTML-Seiten zu vereinfachen. Die x-Koordinate gibt die Strecke nach rechts an, die y-Koordinate die nach unten.

Seitengrößen

Es werden diverse Seitenformate unterstützt. Als Default ist DIN A4 eingestellt. Siehe <document>

Allgemeine Benutzungsschwerpunkte

Template Definition

xxx

Verweise

xxx

Bilder

xxx

Cascading Style Sheets

pisa kann auch ausgezeichnet mit Cascading Style Sheet (CSS) Definitionen wie in HTML umgehen. Folgende Eigenschaften werden unterstützt:

- font-family
- font-size
- color
- line-height
- background-color
- text-decoration: none, underline
- font-weight: bold
- text-align: left, center, right, justify
- text-indent: italic, oblique
- margin
- margin-left
- margin-right
- margin-top
- margin-bottom
- border (und alle Derivate davon, wie z.B. border-right-color)

Leider ist die Unterstützung von CSS noch nicht idiotensicher, z.B. muss bei font-family genau ein Name eingegeben werden und dieser ist auch in der Groß- und Kleinschreibung relevant, wie sonst auch. Dafür können

Größen und andere Maße auch in Zentimetern etc. angegeben werden. padding wird nicht unterstützt, da die reportlab-Library dafür auch keine Funktionalität vorsieht.

Beispiel

Es folgt ein einfaches Beispiel:

```
<style type="text/css">
<!--
h1 {
font-size: 16px;
font-weight: bold;
color: #000000;
margin-bottom: 8px;
margin-top: 0px;
font-family: Helvetica;
}
.test {
color: #444444;
font-family: Helvetica;
font-size: 9px;
}
p {
font-size: 10px;
margin-bottom: 8px;
}
-->
</style>

<h1>Überschrift</h1>
<p>Normaler <span class="test">Text</span></p>
```

Todo

- Verarbeitung von Schriften unempfindlicher machen
- Die bestehenden CSS Möglichkeiten für media:print unterstützen
- Weitere komplexe Definitionsmöglichkeiten erlauben

Referenz

Legende

Jeder Befehl wird einzeln beschrieben. Im Titel steht der Name des Befehls evtl. gefolgt von dem Begriff "(Block)". Dieser Begriff weist darauf hin, dass auch ein schließender Tag erwartet wird, z.B. <p>...</p>.

In den nächsten Zelen folgen die einzelnen Attribute die angewendet werden Können oder Müssen. Die unbedingt erwarteten Attribute sind zusätzlich mit einem Stern versehen. Danach folgt jeweils der Typ des Wertes, die im folgenden einzeln beschrieben werden. In eckigen Klammern wird - wenn vorhanden - der Default-Wert angegeben.

Boolean

Wahr oder Falsch. Falsch wird ausgedrückt durch "0" oder "false", alle anderen Werte werden als Wahr interpretiert.

Integer

Ganze Zahl.

String

Zeichenkette.

File

Ein Dateiname, der sowohl absolut als auch relativ angegeben werden kann.

Font

Name eine installierten oder vorher importierten Schrift. Folgende Schriften sind bei PDF vordefiniert und können daher immer ohne besondere Vorbereitungen verwendet werden:

```
Courier
Courier-Bold
Courier-BoldOblique
Courier-Oblique
Helvetica
Helvetica-Bold
Helvetica-BoldOblique
Helvetica-Oblique
Symbol
Times-Bold
Times-BoldItalic
Times-Italic
Times-Roman
Zapf-Dingbats
```

Color

Farbwert. Wird in der Regel durch einen RGB-Wert in Hexadezimalschreibweise angegeben, z. B. "#ff0000" oder in der abgekürzten Form "#f00". Es ist aber auch möglich gängige Farbnamen zu benutzen, wie z.B. "red", "green", "blue" etc.

Size

Größenangabe. Es wird ein Fließkommazahlenwert angegeben, der dann in Punkte umgerechnet wird. An die Zahl können aber auch Identifikatoren (ohne Leerzeichen!) angehängt werden, die eine Umrechnung ermöglichen. Diese lauten wie folgt:

- "cm" für Zentimeter

- "i", "in" oder "inch" für Inch
- "pt" oder "px" für Punkte

Class

Name einer vorher definierten CSS Klasse.

Style

CSS Style Definitionen, wie z.B. "font-family: Helvetica; font-weight: bold; color: #f00" .

Box

Es werden vier Größenangaben erwartet, die jeweils durch Leerzeichen voneinander getrennt sind. Die ersten zwei Werte geben die obere linke Ecke an und die beiden folgenden die Breite und Höhe. Es ist auch möglich durch negative Vorzeichen bei den letzten beiden Werten einen Abstand relativ zum äußeren Rand anzugeben. Bei einer Frame-Definition bedeutet folgender Wert: "1cm 1cm -1cm -1cm", dass dieser Bereich zu allen Rändern einen 1cm Abstand hat.

Position

Besteht aus zwei durch Leerzeichen getrennten Werten, die eine X/Y-Koordinate definieren.

Choice

Auswahl aus einer vorgegebenen Menge von Werten, die in der Definition ausführlich angegeben sind.

Befehle

Im Folgenden finden Sie eine genaue Beschreibung der einzelnen Befehle.

a (Block)

class	Class [default: 'a']
href	String
name	String
style	Style

Im Attribut "href" steht ein Verweis auf eine andere Website als URL. Es ist aber auch möglich über das Attribut "name" einen Anker zu legen, der dann über "href" angesprungen werden kann.

```
<a href="#test">zur nächsten Seite</a>.  
<nextpage>  
<a name="test"></a>Neue Seite
```

b (Block)

class	Class [default: 'b']
style	Style

Fette Schrift. Es wird empfohlen [](#) zu benutzen.

br

class	Class [default: 'br']
style	Style

Einfacher Zeilenwechsel.

dd (Block)

class	Class [default: 'dd']
style	Style

Beschreibung in einer Definitionsliste. Siehe [<dl>](#).

div (Block)

align	Choice 'left', 'center', 'right', 'justify'
class	Class [default: 'div']
style	Style

Abschnitt als Block. Hauptsächlich zur Verwendung mit CSS.

dl (Block)

class	Class [default: 'dl']
style	Style

Definitionsliste. Die Definitionen erscheinen in einem eigenen Style, die Beschreibungen erscheinen eingerückt.

document (Block)

author	String
duration	Integer
format	Choice 'a0', 'a1', 'a2', 'a3', 'a4', 'a5', 'a6', 'b0', 'b1', 'b2', 'b3', 'b4', 'b5', 'b6', 'letter', 'legal', 'elevenseventeen' [default: 'a4']
fullscreen	Boolean [default: '0']
orientation	Choice 'portrait', 'landscape'
outline	Integer
showoutline	Boolean [default: '0']

subject	String
title	String

Erster Tag eines Dokumentes, der dessen grundlegenden Eigenschaften festlegt. "format" gibt das Format der Dokumentenseiten an. Über "orientation" wird festgelegt, ob die Seite im Hochformat ("portrait") oder Querformat ("landscape") angezeigt wird. "author" und "title" werden in die Dokumenteninformation des PDF eingetragen und sind über entsprechende Funktionen im AcrobatReader auszulesen. Wird "fullscreen" gesetzt, werden im AcrobatReader die Navigationselemente ausgeblendet und evtl. öffnet sich das Dokument sogar bildschirmfüllend (unter Windows abhängig davon, ob das PDF in einem Browser oder direkt im AcrobatReader aufgerufen wird). Wird "showoutline" gesetzt, so wird die Inhaltsstruktur bei der Anzeige des PDF automatisch angezeigt.

drawing

align	Choice 'left', 'center', 'right' [default: 'right']
height	Size
pos*	Position
src*	File
valign	Choice 'top', 'middle', 'bottom' [default: 'bottom']
width	Size

Platziert ein Bild an einer absoluten Stelle im Dokument.

drawline

color	Color [default: '#000000']
from*	Position
to*	Position
width	Size [default: '1']

Zeichnet eine Line von "from" nach "to" mit der Farbe "color". Das folgende Beispiel zeichnet eine Linie von oben links nach unten rechts mit jeweils einem Zentimeter Abstand zum Rand:

```
<drawline from="1cm 1cm" to="-1cm -1cm">
```

drawlines

color	Color [default: '#000000']
coords*	String
width	Size [default: '1']

XXX Wird wahrscheinlich wieder entfernt !!!

Zeichnet mehrere aufeinander folgendes Linien deren Verlauf in "coords" definiert wird. Dabei werden zwei aufeinander folgende Werte als x- und y-Position interpretiert. Folgendes Beispiel zeichnet einen Rahmen um eine Seite mit 1cm Abstand vom Rand:

```
<drawlines coords="1cm 1cm 1cm -1cm -1cm -1cm -1cm 1cm 1cm 1cm">
```

drawpoint

color	Color [default: '#000000']
pos*	Position
width	Size [default: '1']

Zeichnet einen Punkt.

drawstring

align	Choice 'left', 'center', 'right' [default: 'right']
class	Class
color	Color [default: '#000000']
pos*	Position
rotate	Integer [default: '0']
text*	String

valign **Choice** 'top', 'middle', 'bottom' [default: 'bottom']

Schreibt einen String an eine absolute stelle im Dokument.

dt (Block)

class **Class** [default: 'dt']
style **Style**

Definition in einer Definitionsliste. Siehe [<dl>](#).

em (Block)

class **Class** [default: 'em']
style **Style**

Hervorhebung bzw. kursive Schrift.

font (Block)

color **Color**
face **Font**
size **Size**

Schriftarten die dem Acrobat Reader bekannt sind, ohne das sie eingebettet werden müßten, d.h. auch das bei Verwendung dieser Schriften das PDF sehr wenig Speicher benötigt:

fontembed

afm **File**
alias **String**
encoding **String** [default: 'WinAnsiEncoding']
name **String**
pfb **File**
print **Boolean** [default: '0']
ttf **File**

Bindet PostScript-Schriften zur Benutzung im Dokument ein. Benötigt werden die Dateien mit den Endungen ".afm" und ".pfb". Wurde das Attribut "print" angegeben, wird in der PDF Datei eine Debugging-Meldung über den erfolgreichen Import der Schrift und deren Defaultname ausgegeben. Das Einbinden von TrueType-Schriften erfolgt analog über das Attribut "ttf" mit dem Pfad der Schriftdatei.

```
<fontembed ttf="vera.ttf" name="tfont">
<fontembed afm="leerc.afm" pfb="leerc.pfb" name="pfont">
<p><font face="tfont">TrueType, Umlaute: &auml;&ouml;&uuml;</font></p>
<p><font face="pfont">PostScript, Umlaute: &auml;&ouml;&uuml;</font></p>
```

frame

border **Boolean** [default: '0']
box* **Box**
name **String**

Spannt sogenannte Frames auf in die der Text fließen kann. Diese Befehl muss umschlossen sein von dem Befehl "template". Ist ein Frame voll fließt der Text in dem nächsten Frame weiter in der Reihenfolge ihrer Definition. Gibt es keinen nicht benutzten Frame mehr in einem Template wird eine neue Seite erzeugt.

Siehe auch [<template>](#).

h1 (Block)

class **Class** [default: 'h1']
closed **Integer**

outline	String
style	Style

Überschrift 1. Ordnung. Das Attribut "outline" gibt an, unter welchem Namen diese Überschrift im PDF Verzeichnis erreichbar sein soll.

h2 (Block)

class	Class [default: 'h2']
closed	Integer
outline	String
style	Style

Überschrift 2. Ordnung. Siehe [<h1>](#).

h3 (Block)

class	Class [default: 'h3']
closed	Integer
outline	String
style	Style

Überschrift 3. Ordnung. Siehe [<h1>](#).

h4 (Block)

class	Class [default: 'h4']
closed	Integer
outline	String
style	Style

Überschrift 4. Ordnung. Siehe [<h1>](#).

h5 (Block)

class	Class [default: 'h5']
closed	Integer
outline	String
style	Style

Überschrift 5. Ordnung. Siehe [<h1>](#).

h6 (Block)

class	Class [default: 'h6']
closed	Integer
outline	String
style	Style

Überschrift 6. Ordnung. Siehe [<h1>](#).

hr

class	Class [default: 'hr']
color	Color [default: '#000000']
size	Size [default: '1']
style	Style

Ein horizontaler Balken über die ganze Breite des Frames in der Farbe "color".

i (Block)

class	Class [default: 'i']
style	Style

Kursive Schrift. Es wird empfohlen [](#) zu verwenden.

img

align	Choice 'left', 'center', 'right' [default: 'left']
class	Class [default: 'img']
height	Size
src*	File
style	Style
width	Size

Fügt ein Bild in das Dokument ein. Achtung, bei der Berechnung der Größe kann es zu merkwürdigen Ergebnissen kommen. Es sollten nur JPEG Dateien verwendet werden, andere Formate benötigen unter Umständen die Installation des Python Packages PIL.

keepetogether (Block)

Durch <keepetogether> eingeschlossene Textpassagen werden möglichst immer ganz auf einer Seite dargestellt. Beispiel:

```
<keepetogether>
<h1>Überschrift</h1>
<p>Text</p>
</keepetogether>
```

li (Block)

class	Class [default: 'li']
style	Style

Listenelement. Siehe [](#) und [](#).

link

href*	File
rel	String [default: 'stylesheet']
type	String [default: 'text/css']

Lädt eine externe CSS Datei nach.

nextframe

Springt direkt in den nächsten logischen Frame (die Frames folgen in der Reihenfolge ihrer Definition innerhalb von <template> aufeinander). Das direkte Anspringen eines Frames ist (noch) nicht möglich.

Siehe auch [<frame>](#).

nextpage

template	String
----------	---------------

Erzwingt einen Seitenwechsel, auch wenn noch nicht alle Frames benutzt wurden. Hier kann auch direkt festgelegt werden, welches Template die folgenden Seiten benutzen sollen.

nexttemplate

name	String [default: 'default']
------	------------------------------------

Über diesen Befehl wird definiert, welches Template verwendet werden soll, wenn es zum Seitenwechsel kommt. Auf diese Art und Weise ist es auch sehr schön möglich, eine komplexe Startseite zu definieren und dann eine einfachere Folgeseite zu verwenden, wenn der Text so lang ist, dass er eine neue Seite benötigt.

Siehe auch [<template>](#).

ol (Block)

class	Class [default: 'ol']
style	Style
type	Choice '1', 'a', 'A', 'i', 'I' [default: '1']

Geordnete Liste.

p (Block)

align	Choice 'left', 'center', 'right', 'justify'
class	Class [default: 'p']
style	Style

Normaler Textabsatz.

pagenumber

example	String [default: '0']
---------	------------------------------

Fügt an der Stelle die aktuelle Seitennummer ein. In der Phase in der das Layout ausgerechnet wird, kann leider noch nicht die genaue Seitennummer festgestellt werden, deswegen muß in "example" ein Platzhalter für die Länge der Seitennummer angegeben werden. Wenn davon ausgegangen werden kann, daß die Seitennummer einstellig ist, dann sollte example="0" gesetzt werden sonst example="00" und so weiter.

pre (Block)

class	Class [default: 'pre']
style	Style

Vorformatierter Text.

spacer

height*	Size
---------	-------------

Generiert einen neuen Absatz bei dem zusätzlich bestimmt werden kann, wie viel Abstand genau zum nächsten Absatz freigelassen werden soll.

span (Block)

class	Class [default: 'span']
style	Style

Ein Bereich im Fließtext der über CSS gestaltet werden kann.

static (Block)

border	Boolean [default: '0']
box*	Box
name	String

Hierbei handelt es sich um einen speziellen Frame, in den kein Fließtext fließt, sondern der Selbst einen immer wiederkehrenden Teil des Templates definiert, z.B. als Kopf- oder Fußzeile.

Siehe auch [<template>](#).

strong (Block)

class	Class [default: 'strong']
style	Style

Starke Hervorhebung bzw. fette Schrift.

sub (Block)

class	Class [default: 'sub']
style	Style

Tiefgestellte Schrift.

sup (Block)

class	Class [default: 'sup']
style	Style

Siehe [<sup>](#).

super (Block)

class	Class [default: 'super']
style	Style

Hochgestellte Schrift.

table (Block)

align	Choice 'left', 'center', 'right' [default: 'left']
bgcolor	Color
border	Size [default: '0']
bordercolor	Color [default: '#000000']
cellpadding	Size [default: '1']
class	Class [default: 'table']
repeat	Integer [default: '0']
style	Style
valign	Choice 'top', 'bottom', 'middle' [default: 'middle']
width	String

Bei der Implementierung des <table>-Befehls wurde darauf geachtet, dass er möglichst seinem Pendant aus dem HTML Bereich entspricht.

Komplexes Beispiel:

```
<table align="left" grid=#666666>
<tr>
<td width="8.0cm" bgcolor="#dddddd">
<p style=myleft>
<b>Beschreibung</b>
</td>
<td width="1.6cm" bgcolor="#dddddd">
<p style=myright>
<b>Menge</b>
</td>
<td width="3cm" bgcolor="#dddddd">
<p style=myright>
<b>Einzelpreis</b>
</td>
<td width="3cm" bgcolor="#dddddd">
<p style=myright>
<b>Betrag</b>
</td>
</tr>
</table>
```

Ein besonderes Attribut ist "repeat". Wird dieses auf "1" gesetzt, so wird bei einem Seitenwechsel die erste Zeile der Tabelle auf der nächsten Seite wiederholt. Auch der Wert für "align" verhält sich etwas anderes als bei HTML,

denn bei align="left" werden die Tabellen nicht nebeneinander dargestellt, sondern immer untereinander.

td (Block)

align	Choice 'left', 'center', 'right', 'justify'
bgcolor	Color
border	Size
bordercolor	Color [default: '#000000']
class	Class [default: 'td']
colspan	Integer
rowspan	Integer
style	Style
valign	Choice 'top', 'bottom', 'middle'
width	String

Tabellenspalte. Siehe auch [<table>](#).

template (Block)

background	File
format	Choice 'a0', 'a1', 'a2', 'a3', 'a4', 'a5', 'a6', 'b0', 'b1', 'b2', 'b3', 'b4', 'b5', 'b6', 'letter', 'legal', 'elevenseventeen' [default: 'a4']
name	String [default: 'default']
orientation	Choice 'portrait', 'landscape'

Definiert ein Seitentemplate, das verschiedene Frames enthält. Das erste vordefinierte Template heißt "default" und besteht aus einem einzelnen Frame. Dieses Template kann überschrieben werden, indem sie ein Template mit dem Namen "default" anlegen oder das Attribut "name" einfach weg lassen. Achtung, sollten mehrere Templates mit dem gleichen Namen definiert werden, so wird nur das zuletzt definierte tatsächlich benutzt!

Für jedes einzelne Template kann auch eine andere Seitengröße und -orientierung festgelegt werden. Wird das nicht gemacht, dann greifen die Einstellungen aus <document>.

Über das Attribut "background" kann eine PDF Datei angegeben werden, die auf den Seiten die das Template benutzt wird in den Hintergrund gelegt wird. Aus technischen Gründen geht bei Benutzung dieses Features eine evtl. angelegtes "Outline" (siehe [h1](#) etc.) dadurch verloren.

Hier ein kleines Beispiel:

```
<template name="default">
<frame name="head"
box="17,3cm 2.8cm 3.3cm 20cm">
<frame name="address"
box="2.7cm 5.0cm 8.57cm 4.00cm">
<frame name="body"
box="3.5cm 10.5cm 13.9cm 17.5cm">
</template>

<template name="sub">
<frame name="c"
box="3.5cm 3.5cm 13.9cm 17.5cm">
</template>
```

Innerhalb von Template sind die Definitionen von [<frame>](#) und [<static>](#) vorgesehen.

Des weiteren kann die Templatedefinition grafische Anweisungen, wie [<drawline>](#) etc. enthalten, die dann bei jeder neuen Seite die das Template benutzt vor dem Füllen der Seite ausgeführt werden.

th (Block)

align	Choice 'left', 'center', 'right', 'justify'
bgcolor	Color
border	Size
bordercolor	Color [default: '#000000']

class	Class [default: 'th']
colspan	Integer
rowspan	Integer
style	Style
valign	Choice 'top', 'bottom', 'middle'
width	String

Tabellenkopf. Siehe auch [<table>](#).

tr (Block)

bgcolor	Color
border	Size
bordercolor	Color [default: '#000000']
class	Class [default: 'tr']
style	Style
valign	Choice 'top', 'bottom', 'middle'

Tabellenzeile. Siehe auch [<table>](#).

u (Block)

class	Class [default: 'u']
style	Style

Unterstrichen.

ul (Block)

class	Class [default: 'ul']
style	Style
type	Choice 'circle', 'disk', 'square' [default: 'disk']

Ungeordnete Liste.

version

Gibt die aktuelle Versionsnummer von PISA aus.

Bekannte Fehler

- Rahmen werden nur in Tabellen unterstützt

Ausblick

Folgende Features sollten noch kommen:

- Weitere Grafikfunktionen
- Ausbau der CSS Funktionalität
- SVG Einbindung
- Bessere Listen
- Bessere Preformatted Bereiche
- Seitennummern ohne Example und Gesamtseitenzahl
- Inhaltsverzeichnis automatisch generieren
- Aktuelles Datum
- Dokumententitel
- Outlines

Impressum

spirito GmbH
Grünstraße 6
47051 Duisburg
Telefon: +49.203.3187777
E-Mail: info@spirito.de
Internet: www.spirito.de

Vertretungsberechtigte Geschäftsführer:
Martin Gerardi und Dirk Holtwick

Registergericht:
Amtsgericht Duisburg

Registernummer:
HR 9304

Haftungshinweis:
Trotz sorgfältiger inhaltlicher Kontrolle übernehmen wir keine Haftung für die Inhalte externer Links. Für den Inhalt der verlinkten Seiten sind ausschließlich deren Betreiber verantwortlich.

Q Public License v1.0

Copyright © 1999 Trolltech AS, Norway.

Everyone is permitted to copy and distribute this license document.

The intent of this license is to establish freedom to share and change the software regulated by this license under the open source model.

This license applies to any software containing a notice placed by the copyright holder saying that it may be distributed under the terms of the Q Public License version 1.0. Such software is herein referred to as the Software. This license covers modification and distribution of the Software, use of third-party application programs based on the Software, and development of free software which uses the Software.

Granted Rights

1. You are granted the non-exclusive rights set forth in this license provided you agree to and comply with any and all conditions in this license. Whole or partial distribution of the Software, or software items that link with the Software, in any form signifies acceptance of this license.
2. You may copy and distribute the Software in unmodified form provided that the entire package, including - but not restricted to - copyright, trademark notices and disclaimers, as released by the initial developer of the Software, is distributed.
3. You may make modifications to the Software and distribute your modifications, in a form that is separate from the Software, such as patches. The following restrictions apply to modifications:
 - a. Modifications must not alter or remove any copyright notices in the Software.
 - b. When modifications to the Software are released under this license, a non-exclusive royalty-free right is granted to the initial developer of the Software to distribute your modification in future versions of the Software provided such versions remain available under these terms in addition to any other license(s) of the initial developer.
4. You may distribute machine-executable forms of the Software or machine-executable forms of modified versions of the Software, provided that you meet these restrictions:
 - a. You must include this license document in the distribution.
 - b. You must ensure that all recipients of the machine-executable forms are also able to receive the complete machine-readable source code to the distributed Software, including all modifications, without any charge beyond the costs of data transfer, and place prominent notices in the distribution explaining this.
 - c. You must ensure that all modifications included in the machine-executable forms are available under the terms of this license.
5. You may use the original or modified versions of the Software to compile, link and run application programs legally developed by you or by others.
6. You may develop application programs, reusable components and other software items that link with the original or modified versions of the Software. These items, when distributed, are subject to the following requirements:
 - a. You must ensure that all recipients of machine-executable forms of these items are also able to receive and use the complete machine-readable source code to the items without any charge beyond the costs of data transfer.
 - b. You must explicitly license all recipients of your items to use and re-distribute original and modified versions of the items in both machine-executable and source code forms. The recipients must be able to do so without any charges whatsoever, and they must be able to re-distribute to anyone they choose.
 - c. If the items are not available to the general public, and the initial developer of the Software requests a copy of the items, then you must supply one.

Limitations of Liability

In no event shall the initial developers or copyright holders be liable for any damages whatsoever, including - but

not restricted to - lost revenue or profits or other direct, indirect, special, incidental or consequential damages, even if they have been advised of the possibility of such damages, except to the extent invariable law, if any, provides otherwise.

No Warranty

The Software and this license document are provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Choice of Law

This license is governed by the Laws of Norway. Disputes shall be settled by Oslo City Court.