

Ce « Guide de l'Utilisateur » décrit le mode d'emploi d'un outil libre (cf. « **License** » ci-dessous), gratuit, et portable, de création de livres électroniques aux formats ePUB et PDF à partir de documents HTML rédigés avec *OpenOffice Writer*. Fonctionnant sur le principe d'une conversion préalable des documents au format XHTML 1.1 accompagnés d'une feuille de style CSS 2.1 partagée, préalablement à l'assemblage du livre électronique au format ePUB et/ou à la composition de sa version imprimable au format PDF, il offre, en mode dit « expert », la possibilité de réutiliser les documents convertis après y avoir introduit des améliorations de style qu'*OpenOffice Writer* ne permet pas.

Les caractéristiques principales des documents qu'il produit sont :

🔗 Conversion HTML vers XHTML :

- 🔗 Des composants XHTML 1.1 compatibles XHTML 5, optimisés et lisibles.
- 🔗 Des styles CSS 2.1 unifiés reproduisant la présentation originale des documents.

🔗 Assemblage du livre électronique au format ePUB :

- 🔗 Un fichier ePUB conforme aux spécifications ePUB 2.0.1 de l'IDPF.
- 🔗 Des composants XHTML éventuellement réorganisés pour a) ne pas dépasser une taille maximale, et/ou b) toujours commencer par un titre de même niveau. Le fichier ePUB est alors compatible avec les dispositifs de lecture basés sur *Adobe® Digital Editions*.
- 🔗 Des notes de bas de page regroupées, et le cas échéant, reportées en fin de sections de même niveau ou de document.
- 🔗 Des graphiques aux formats JPeG, PNG ou SVG.
- 🔗 La possibilité d'inclure des polices de caractères dans le fichier ePUB.
- 🔗 La possibilité de générer des polices de caractères ne contenant que les caractères utilisés, avant inclusion dans le fichier ePUB¹.

🔗 Composition du document PDF :

- 🔗 Un fichier PDF 1.6 de qualité typographique professionnelle composé avec le logiciel ConT_EXt.
- 🔗 Une organisation et une présentation enrichie avec pages liminaires, pages post-liminaires, annexes, index, bibliographie, en-têtes et pieds de page, etc...
- 🔗 Des graphiques aux formats JPeG ou PNG.

🔗 En mode « expert », à partir de documents XHTML et feuilles de style CSS :

- 🔗 Une gestion fine des polices de caractères : les propriétés font-stretch et font-size-adjust introduites par le module CSS Font Module Level 3 sont reconnues et permettent respectivement de gérer la chasse et d'ajuster entre elles la taille des polices de caractères hétérogènes pour qu'elles paraissent visuellement homogènes. La feuille de style CSS produite reste, elle, compatible avec la norme CSS 2.1.
- 🔗 Le changement automatique de famille de polices de caractères selon l'alphabet utilisé, en fonction des familles définies par la propriétés font-family.

Pré-requis

L'outil ne requiert que la distribution ConT_EXt Minimals du logiciel de publication ConT_EXt et l'utilitaire Info-Zip. Ceux-ci sont disponibles pour les systèmes Windows, Mac OS/X et nombre de systèmes de type Unix. Avec ces derniers, l'éditeur de polices de caractères FontForge permettra aussi, s'il est installé, de diminuer l'encombrement des polices de caractères intégrées au fichier ePUB.

¹ Sur les systèmes d'exploitation de type Unix uniquement.

This is the “User’s Guide” of an open-source (cf. “[License](#)” below), free, and OS-independent, ePUB and PDF electronic books creation tool from *OpenOffice Writer* HTML documents. Using an intermediary HTML to XHTML 1.1 and CSS 2.1 conversion step, before assembling components into an ePUB file and/or composing the PDF printable document, its “expert” mode provides for reusing stylistically enhanced—when *OpenOffice Writer* alone cannot manage the desired results—versions of these intermediary produced files as input.

Its main features are:

🔗 HTML to XHTML conversion:

- 🔗 Cleansened and nicely formatted XHTML 5 compatible XHTML 1.1 components.
- 🔗 Shared CSS 2.1 styles retaining the original components’ layouts.

🔗 ePUB electronic book assembly:

- 🔗 A fully-compliant to the IDPF specifications ePUB 2.0.1 file
- 🔗 An *Adobe® Digital Editions SDK* compatible ePUB file: XHTML components may be restructured into maximum sized files, and/or each starting at the same title level.
- 🔗 Footnotes may be gathered at the end of sections starting with the same title level or at the end of the document.
- 🔗 JPeG, PNG or SVG graphics.
- 🔗 Optional embedded fonts in the ePUB file.
- 🔗 Font subsetting prior to embedding².

🔗 PDF printable document composition:

- 🔗 A professional typographic quality PDF 1.6 document composed with ConT_EXt.
- 🔗 A structured and enhanced (front and back matter, appendices, table of contents, index, bibliography, page headers and footers, etc.) PDF document.
- 🔗 JpeG or PNG graphics.

🔗 “expert” mode, when using XHTML and CSS files as input:

- 🔗 CSS Font Module Level 3 (partial) compatibility brings in the font-stretch and font-size-adjust properties for finer font management.
- 🔗 Automatic script dependent font family switching, based on the content of the font-family property.

Requirements

The ConT_EXt Minimals distribution of the ConT_EXt publication software and the Info-Zip tool are required. These are available to Windows, Mac OS/X, and many Unix-like operating systems. With these last OSes, the FontForge font editor provides, if installed, for subsetting fonts before embedding them into the ePUB file.

² On Unix Operating Systems only.

Copyright © Frédéric Haby, 2011.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

The Software is provided “as is”, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. in no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software

Table des matières

Résumé	1
Pré-requis	1
Abstract	3
Requirements	3
License	4
1 Des polices, des polices et encore des polices	7
Remarques préliminaires	7
Changement de famille de polices de caractères selon les « alphabets »	9
Modification de la chasse de la police courante	11
Police de caractères requise et polices de caractères utilisées	12
Ajustement de la taille des polices de caractères	13
La représentation des caractères Unicode	14
Exemples	14
1.1 More	19
Contents	19
Remerciements	21
Aknowledgements	23
Index	25
Bibliographie	27

Des polices, des polices et encore des polices

Les principes généraux d'affectation des polices de caractères au texte ont été décrits au chapitre « » ; tandis que le recours aux polices de caractères « génériques » a été évoqué dans le chapitre « ». Ce chapitre illustre l'ensemble des points mentionnés dans ces deux chapitres et explicite dans chaque cas comment le fonctionnement interne de whatever effectue la conversion de styles CSS « compatibles » CSS Font Module Level 3, en styles CSS 2.1 et marques XHTML produisant le même résultat final, après a) élimination des propriétés font-stretch et font-size-adjust b) substitution de mots-clés aux familles de polices de caractères « génériques »—et inversement, selon le cas c) réduction des listes de familles de la propriété font-family à leur plus simple expression possible.

Remarques préliminaires

Les familles de polices de caractères « génériques » utilisées

Tous les exemples donnés se basent sur la définition des familles de polices de caractère « génériques » utilisée pour la composition de ce document, c'est-à-dire :

<i>Famille de polices de caractères « générique »</i>	<i>Famille de polices de caractères associées lors de la composition</i>
serif	Alfios
sans-serif	Aurelis ADF N°2 Std
monospace	Verily Serif Mono
cursive	Aurelis ADF Script N°2 Std
decorative	Gardiner
fantasy	Akkadian

La contrainte de base

La conversion de styles CSS « compatibles » CSS Font Module Level 3 doit produire des styles CSS 2.1 dont la propriété font-family :

1. Mentionne au plus une seule famille de polices de caractères non « générique ».
2. Mentionne toujours, le cas échéant, cette famille de polices de caractères non « générique » en tête de la liste.
3. Mentionne toujours les trois familles de polices de caractères « génériques » de base, c'est-à-dire : serif, sans-serif, et monospace.

Ce principe garantit dans tous les cas l'affichage par un dispositif de lecture donné, de tous les caractères dont il dispose, même son éventail de familles de polices de caractères se réduit à sa plus simple expression—une et une seule des trois polices de caractères génériques de base.

Inversement, il implique qu'il n'est pas possible de forcer l'affichage du texte dans l'une des trois polices de caractères « réelles » affectées à l'une de ces polices de caractères « génériques » de base.

Ceci ne vaut, évidemment, que pour le document ePUB composé. Dans la version PDF, ce sont les polices « réelles » demandées qui sont employés.

Le catalogue des polices de caractères

Avant de détailler le processus d'affectation des polices de caractères, des précisions sur la manière dont ces dernières sont cataloguées lors de leur installation sont indispensables à la compréhension de ce processus.

whatever accède aux différents éléments qui décrivent la police de caractères dans le fichier correspondant et en déduit, éventuellement après recoupement de plusieurs des informations obtenues, des valeurs pour les différentes propriétés, font-style, font-weight, font-variant et font-stretch, de cette police de caractères, ainsi que son nom de famille font-family. Ce processus permet généralement de regrouper correctement toutes les polices de caractères d'une même famille au sein d'une seule famille, quant bien même ce ne serait pas trivialement le cas au vu de données individuelles présentes dans le fichier.

Dans le même temps, il en extrait aussi le type de police dont il s'agit et le rapport entre hauteurs des minuscules et des majuscules, ou le calcule; et enfin il compile aussi la liste des caractères qu'elle contient. Toutes ces informations sont ensuite mises sous forme d'une règle CSS @font-face. C'est celle-ci qui sera ensuite utilisée dans toutes les opérations sur les polices de caractères.

Par exemple, la famille de polices de caractères « cursive » utilisée dans ce document est une police en italiques, qui se décline en trois chasses : condensée, normale et expansée. La variante condensée est cataloguée comme suit :

```
@font-face {
  font-family:      "Aurelis ADF Script No2 Std", cursive;
  font-stretch:     condensed;
  font-weight:      normal;
  font-style:       italic;
  font-variant:     normal;
  font-size-adjust: 0.42;
  src: local("Aurelis ADF Script No2 Std Cd It"),
      url("../fonts/opentype/AurelisADFScriptNo2Std-CdIta.otf") format("opentype");
  unicode-range: U+20-7E, U+A0-FF, U+131, U+141-142, U+152-153,
      U+160-161, U+178, U+17D-17F, U+192, U+2C6-2C7, U+2D8-2DD, U+3C0,
      U+2000-200A, U+2010-2016, U+2018-2026, U+2030, U+2039-203A, U+2044,
      U+20AC, U+2113, U+2122, U+2126, U+212E, U+2202, U+2206, U+220F,
      U+2211-2212, U+221A, U+221E, U+222B, U+2248, U+2260, U+2264-2265,
      U+25CA, U+E674, U+E6A4-E6A5, U+F6C3, U+FB00-FB06;
}
```

Notons au passage que cette définition n'est pas utilisable telle quelle dans la feuille de style CSS 2.1 d'un fichier ePUB. De fait, elle lui sera ultérieurement adaptée, par exemple—l'URL associée dépendant des options choisies lors de la composition—en :

```
@font-face {
  font-family:      "Aurelis ADF Script No2 Std Condensed";
  font-weight:      normal;
  font-style:       italic;
  font-variant:     normal;
  src: local("Aurelis ADF Script No2 Std Cd It"),
      url("../fonts/opentype/AurelisADFScriptNo2Std-CdIta.otf");
}
```

Elle devra donc être invoquée, dans les fichiers XHTML du document ePUB, par une séquence du type :

```
<p>
  Texte en police principale et <span class="style001"><i>texte en police cursive
```


condensée

et une règle :

```
span.style001 {
  font-family:      "Aurelis ADF Script No2 Std Condensed", serif, sans-serif, monospace;
}
```

dans la feuille de style CSS 2.1, règle qui, en outre, respecte la contrainte de base explicitée plus haut.

Changement de famille de polices de caractères selon les « alphabets »

Dans un premier temps, whatever modifie les listes de familles de polices caractères décrites dans les propriétés font-family des fichiers source, de la façon suivante :

- ❧ Les noms de famille de polices de caractères « non génériques » sont conservés.
- ❧ Le premier identifiant d'une famille « générique » « de base » et ceux des familles génériques qui ne sont pas « de base » sont remplacés par les noms des familles correspondantes.
- ❧ Les autres identifiants de familles « génériques » « de base » sont conservés.
- ❧ Les autres noms de familles « génériques » « de base » sont remplacés par les identifiants correspondants.
- ❧ Les identifiants de familles « génériques » « de base » dont le nom ou l'identifiant n'y figure pas encore sont ajoutés à la liste.

Ce sont les familles de polices de caractères explicitement mentionnées—c'est-à-dire à l'exception des identifiants de familles « génériques »—dans la liste ainsi modifiée—qui sont utilisées pour déterminer dans quelle police écrire chaque mot du texte auquel la propriété font-family s'applique—selon l'inventaire du contenu de chacune de ces polices, tel que décrit par sa propriété unicode-range dans le catalogue.

Pour chaque « mot » du texte—à comprendre comme a) partie d'un texte ne comportant pas ni espacement, ni ponctuation, b) caractère d'espacement ou de ponctuation—whatever teste l'utilisation de chacune de ces familles de polices de caractères, dans l'ordre indiqué, et retient la première de celle qui permet d'en représenter le plus de caractères, sauf s'il s'agit de ponctuations, la préférence étant alors donnée, si possible, à la famille de polices de caractères en cours d'utilisation.

Voici quelques exemples de construction de cette représentation interne à whatever de la liste des familles de polices de caractères requises, qui utilisent les définitions de familles « génériques » mentionnées dans la sous-section « **Les familles de polices de caractères « génériques » utilisées** » :

- ❧ « Verily Serif Mono » devient « Verily Serif Mono, serif, sans-serif ».
- ❧ « Verily Serif Mono, sans-serif » devient « Verily Serif Mono, sans-serif, serif ».
- ❧ « Verily Serif Mono, Aurelis ADF No2 Std » devient « Verily Serif Mono, Aurelis ADF No2 Std, serif ».
- ❧ « Bitstream Vera Sans Mono » devient « Bitstream Vera Sans Mono, Alfios, sans-serif, monospace ».
- ❧ « Bitstream Vera Sans Mono, sans-serif » et « Bitstream Vera Sans Mono, Aurelis ADF No2 Std » deviennent « Bitstream Vera Sans Mono, Aurelis ADF No2 Std, serif, monospace ».
- ❧ « Bitstream Vera Sans Mono, monospace » et « Bitstream Vera Sans Mono, Verily Serif Mono » deviennent « Bitstream Vera Sans, Verily Serif Mono, serif, serif, sans-serif ».
- ❧ « decorative, fantasy, sans-serif » devient « Gardiner, Akkadian, Aurelis ADF No2 Std, serif, monospace ».

- 🔗 « sans-serif, decorative, fantasy » devient « Aurelis ADF No2 Std, Gardiner, Akkadian, serif, monospace ».
- 🔗 « decorative, fantasy » et « decorative, fantasy, serif » deviennent « Gardiner, Akkadian, Alfios, serif, monospace ».
- 🔗 « cursive » devient « Aurelis ADF Script No2 Std, Alfios, sans-serif, monospace ».

Les deux derniers exemples sont illustrés plus loin dans ce chapitre : le premier par un paragraphe utilisant une police de caractères cursive, le second avec un paragraphe mélangeant français, hiéroglyphes égyptiens et écriture cunéiforme.

Une fois les familles de polices réellement employées dans les différentes portions du texte connues, whatever calcule les valeurs des propriétés font-family qu'il utilisera dans les styles correspondants de la feuille de style CSS 2.1 :

- 🔗 En conservant la police de caractères retenue pour écrire la portion de texte à laquelle s'applique le style. Si c'est une des polices « génériques » de base, le mot-clef correspondant lui est substitué.
- 🔗 En ajoutant ensuite à la liste les identifiant des autres polices de caractères « génériques », dans le même ordre que dans la représentation interne.

Ce qui peut encore être illustrés par nos exemples précédents :

- 🔗 « Verily Serif Mono » devient toujours « monospace, serif, sans-serif ».
- 🔗 « Verily Serif Mono, sans-serif » devient toujours « monospace, sans-serif, serif ».
- 🔗 « Verily Serif Mono, Aurelis ADF No2 Std » devient « monospace, sans-serif, serif » ou « sans-serif, serif, monospace ».
- 🔗 « Bitstream Vera Sans Mono, sans-serif » et « Bitstream Vera Sans Mono, Aurelis ADF No2 Std » deviennent « Bitstream Vera Sans Mono, sans-serif, serif, monospace » ou « sans-serif, serif, monospace ».
- 🔗 « Bitstream Vera Sans Mono, monospace » et « Bitstream Vera Sans Mono, Verily Serif Mono » deviennent « Bitstream Vera Sans, monospace, serif, sans-serif » ou « monospace, serif, sans-serif ».
- 🔗 « decorative, fantasy, sans-serif » et « sans-serif, decorative, fantasy » deviennent « Gardiner, sans-serif, serif, monospace », « Akkadian, sans-serif, serif, monospace » ou « sans-serif, serif, monospace ».
- 🔗 « decorative, fantasy » et « decorative, fantasy, serif » deviennent « Gardiner, serif, sans-serif, monospace », « Akkadian, serif, sans-serif, monospace » ou « serif, sans-serif, monospace ».
- 🔗 « cursive » devient « Aurelis ADF Script No2 Std, serif, sans-serif, monospace » ou « serif, sans-serif, monospace ».

En procédant de la sorte, les changements de polices de caractères qui s'imposent sont bien introduits en respectant au mieux les indications des styles d'origine, tandis que la contrainte énoncée par la sous-section « **La contrainte de base** » s'applique effectivement aux styles calculés par whatever.

À propos de ponctuation

Sont considérés comme caractères de ponctuation : l'apostrophe, la virgule, le point-virgule et le point-virgule réfléchi, le point, les « deux-points », les points de suspension, le point d'interrogation, le point d'exclamation, le point d'exclamation renversé et le double point d'exclamation, les guillemets sous toutes leurs formes, les parenthèses, les crochets, les accolades, les tirets, la perluette, l'astérisque, le tilde, l'obèle et la double obèle, l'astérisme, les barres verticales et obliques, le symbole « paragraphe », le pied-de-mouche, le symbole « copyright » et le symbole « marque déposée ».

Modification de la chasse de la police courante

La modification de la chasse d'une police dans le cours du texte se traduit par un changement de famille de police de caractères dans les fichiers XHTML compilés par whatever. Comme rien ne les distingue entre elles, hormis la propriété `font-stretch` que la norme CSS 2.1 ne reconnaît pas, il est faut introduire artificiellement une différence sans altérer la signification de propriétés qui définissent une police de caractères : cela ne peut se faire qu'avec la propriété `font-family`, dont la valeur n'est en fin de compte, qu'une convention que l'on utilise pour regrouper entre elles plusieurs polices de caractères. En concaténant la chasse de la police au nom de la famille, ces nouvelles « familles » de police de caractères, qui n'ont pas d'autre existence que celle que leur donne temporairement whatever au sein du projet, restent compatibles avec l'utilisation de marques, telles que `` ou `<i>`, qui modifient les valeurs de leurs autres propriétés. Par exemple :

```
p.style001 {
    font-family:      "Aurelis ADF Script No2 Std";
    font-style:       italic;
}
span.style001 {
    font-stretch:     narrower;
}
span.style002 {
    font-stretch:     wider;
}
...
<p class="style001">
    Du texte normal, <span class="style001">condensé</span>, et
    <span class="style002">expansé</span>.</p>
```

deviendra :

```
@font-face {
    font-family:      "Aurelis ADF Script No2 Std Condensed";
    font-weight:      normal;
    font-style:       italic;
    font-variant:     normal;
    src: local("Aurelis ADF Script No2 Std Cd It"),
        url("../fonts/opentype/AurelisADFScriptNo2Std-CdIta.otf");
}
@font-face {
    font-family:      "Aurelis ADF Script No2 Std Extended";
    font-weight:      normal;
    font-style:       italic;
    font-variant:     normal;
    src: local("Aurelis ADF Script No2 Std Cd Ext"),
        url("../fonts/opentype/AurelisADFScriptNo2Std-CdIta.otf");
}
p.style001 {
    font-family:      "Aurelis ADF Script No2 Std", serif, sans-serif, monospace;
    font-style:       italic;
}
```

```
span.style001 {
  font-family:      "Aurelis ADF Script No2 Std Condensed", serif, sans-serif, monospace;
}
span.style002 {
  font-family:      "Aurelis ADF Script No2 Std Expanded", serif, sans-serif, monospace;
}
...
<p class="style001">
  Du texte normal, <span class="style001">condensé</span>, et
  <span class="style002">expansé</span>.</p>
```

Notons aussi que le code compilé produira de nouveau le même résultat, s'il est utilisé ultérieurement comme source pour whatever. C'est d'ailleurs une des caractéristiques générales—c'est-à-dire qu'elle vaut pour tout le code XHTML et CSS 2.1 qu'il produit—de l'outil, qui n'a simplement pas été mentionnée jusqu'ici. On pourra, par exemple, vérifier que c'est aussi le cas des exemples qui seront donnés dans la suite de ce chapitre.

Police de caractères requise et polices de caractères utilisées

Selon le contexte, défini par la succession de balises HTML et styles CSS qui s'appliquent à une portion de texte, le catalogue ne répertorie pas toujours une police de caractères qui possède exactement les mêmes caractéristiques, définies par les cinq propriétés `font-family`, `font-stretch`, `font-style`, `font-variant` et `font-weight`, que la police de caractère requise. whatever lui substituera alors une police de caractères approchante, obligatoirement de la même famille, en cherchant à respecter autant que faire se peut les valeurs des quatre autres propriétés, et ce en priorité dans l'ordre mentionné ci-dessus. Ainsi, si le contexte réclame une police condensée de style normal, une police condensée mais de style italique sera préférée à une police de chasse et style normaux.

De sorte que l'exemple donné initialement dans la sous-section « **Le catalogue des polices de caractères** » peut aussi s'écrire—notons que la police de caractère requise est définie dans le catalogue et n'a pas donc pas besoin de l'être à nouveau localement—dans les fichiers source :

```
span.style001 {
  font-family:      "Aurelis ADF Script No2 Std";
  font-stretch:      condensed;
}
...
<p>
  Texte en police principale et <span class="style001">texte en police cursive
  condensée</span>.</p>
```

voire même, dans le seul fichier HTML :

```
<p>
  Texte en police principale et <span style="font-family: Aurelis ADF Script No2 Std;
  font-stretch: condensed;">texte en police cursive condensée</span>.</p>
```

La seule différence étant que whatever compilera alors une règle CSS prenant en considération le passage de roman en italiques. On aura dans ce cas, dans les fichiers compilés :

```
@font-face {
  font-family:      "Aurelis ADF Script No2 Std Condensed";
```

```

font-weight:      normal;
font-style:       italic;
font-variant:     normal;
src: local("Aurelis ADF Script No2 Std Cd It"),
    url("../fonts/opentype/AurelisADFScripNo2Std-CdIta.otf");
}
span.style001 {
font-family:      "Aurelis ADF Script No2 Std Condensed", serif, sans-serif, monospace;
font-style:       italic;
}
...
<p>
Texte en police principale et <span class="style001">texte en police cursive
condensée</span>.</p>

```

Ajustement de la taille des polices de caractères

Cette opération s'effectue de façon très simple. Si nous avons, par exemple, dans le composant source, les styles et le texte suivant :

```

p.style001 {
font-family:      Alfios;
font-size:        1.000em;
font-size-ajust:  0.37;
}
code.style001 {
font-family:      "Verily Serif Mono";
}

<p class="style001">
<code class="style001">whatever</code>compilera :</p>

```

whatever compilera :

```

p.style001 {
font-family:      serif, sans-serif, monospace;
font-size:        1.000em;
}
code.style001 {
font-family:      monospace, serif, sans-serif;
font-size:        0.712em;
}

<p class="style001">
<code class="style001">whatever</code>compilera :</p>

```

En effet, 0.37 est la valeur du rapport hauteur de minuscules / hauteur de majuscules de la famille de polices de caractères « Alfios », la taille de celle-ci reste donc inchangée, tandis que ce rapport vaut 0.52 pour la famille « Verily Serif Mono » dont la taille devient alors $1\text{em} * 0.37 / 0.52 = 0.712\text{em}$.

Remarquons que c'est le mécanisme d'héritage des propriétés `font-size` et `font-size-adjust` qui a ici été mis en œuvre, c'est-à-dire que—bien que ce ne soit pas un d'usage courant—le style :

```
code.style001 {
  font-family:      "Verily Serif Mono";
  font-size:        1.000em;
  font-size-adjust: 0.37;
}
```

produirait le même résultat.

La représentation des caractères Unicode

ISO-8859-1 et UTF-8

OpenOffice Writer génère des fichiers HTML codés en US-ASCII, sous-ensemble de la norme ISO-8859-1. L'alphabet US-ASCII ne comporte que 96 caractères imprimables. Les autres caractères sont représentés soit par des entités HTML, soit par leur valeur numérique, ou « point de code » Unicode. Ainsi, un « à » peut s'écrire « à » ou « à ». La norme UTF-8 pour sa part, permet de représenter, sans recourir à des tables de transcodage—ce que sont de fait les entités HTML—la totalité des caractères du standard Unicode, et au-delà. Les spécifications ePUB 2.0.1 et 3.0 imposent l'utilisation de cette dernière norme, mais, dans les faits, les dispositifs de lecture ne la comprennent pas tous.

Pour ces différentes raisons, *whatever* accepte les fichiers HTML codés en ISO-8859-1 et ceux codés en UTF-8, et compile a) des fichiers ePUB codés, par défaut en UTF-8, mais aussi, en option, en ISO-8859-1, b) des composants XHTML, hors fichiers ePUB, codés en ISO-8859-1.

Les paires de substitution

Une difficulté supplémentaire surgit du fait de la possibilité de coder de deux manières différentes, en ISO-8859-1, les points de code Unicode au-delà de « U+FFFF », c'est-à-dire dont la valeur numérique nécessite plus de quatre octets pour être représentée en binaire. Par exemple, le point de codage « U+2A6D6 » peut s'écrire « 񪛖 » en valeur décimale ou en le remplaçant par la paire de points de code « U+0xD869 » et « U+0xDE6 », soit en valeur décimale « � � » si l'on utilise les paires de substitutions. C'est cette dernière représentation qu'utilise *OpenOffice Writer*.

whatever accepte donc et les valeurs décimales et les paires de substitution dans ses fichiers source et compile soit des valeurs décimales soit des paires de substitution dans les composants XHTML—hors fichiers ePUB—codés en ISO-8859-1. Dans le second cas, les fichiers produits sont techniquement des fichiers HTML, puisque qu'ils ne commencent plus par une balise « <?xml ?> ».

Exemples

Changements « automatiques » de familles de polices de caractères

Le paragraphe ci-dessous contient du texte en écriture latine, hiéroglyphes égyptiens et écriture cunéiforme. Voici le code XHTML et CSS 3.0 source—où, pour illustrer les deux possibilités, les hiéroglyphes sont notés avec des « paires de substitution » et les caractères cunéiformes avec les valeurs décimales de leurs points de code :

```
p.style001 {
  font-family:      serif, decorative, fantasy;
}
...
```

```
<p class="style001">
Voici des hiéroglyphes
&#56204;&#56320;&#56204;&#56321;&#56204;&#56322;&#56320;
et &#73728;&#73729;&#73730; du cunéiforme.</p>
```

Le code XHTML et CSS 2.1 généré ressemblera alors à ce qui suit :

```
p.style001 {
    font-family:      serif, sans-serif, monospace;
}
span.style001 {
    font-family:      Aegyptus, serif, sans-serif, monospace;
}
span.style002 {
    font-family:      Akkadian, serif, sans-serif, monospace;
}

...
<p class="style001">
Voici des hi&eacute;rogl&eacute;phes <span
class="style001">&#995328;&#995329;&#995330;</span> et <span
class="style002">&#73728;&#73729;&#73730;</span> du
cun&eacute;iforme.</p>
```

Ce que le dispositif de lecture affichera :

Voici des hiéroglyphes  et  du cunéiforme.

Changement de chasse de la police courante

Le paragraphe qui suit utilise les trois polices de caractères de la famille « Aurelis ADF Script N°2 Std » dont les chasses diffèrent. Voici d'abord le code XHTML et CSS 3.0 source :

```
p.style001 {
    font-family:      serif, sans-serif, monospace;
}
span.style001 {
    font-family:      cursive, serif, sans-serif, monospace;
}
span.style002 {
    font-family:      cursive, serif, sans-serif, monospace;
    font-stretch:      narrower;
    font-style:        italic;
}
span.style003 {
    font-family:      cursive, serif, sans-serif, monospace;
    font-stretch:      wider;
    font-style:        italic;
}

...
<p class="style001">
```

Les polices de caractères de la famille « Aurelis ADF Script N°2 Std » existent en version Aurelis ADF Script N°2 Std normale, Aurelis ADF Script N°2 Std condensée et Aurelis ADF Script N°2 Std expansée.

Le code XHTML et CSS 2.1 généré ressemblera alors à ce qui suit (les directives @font-face sont celles qui seraient générés si les polices de la famille « Aurelis ADF Script N°2 Std » étaient intégrées au fichier ePUB) :

```
p.style001 {
    font-family:      serif, sans-serif, monospace;
}
span.style001 {
    font-family:      "Aurelis ADF Script No2 Std", serif, sans-serif, monospace;
}
span.style002 {
    font-family:      "Aurelis ADF Script No2 Std Condensed", serif, sans-serif, monospace;
}
span.style003 {
    font-family:      "Aurelis ADF Script No2 Std Expanded", serif, sans-serif, monospace;
}
@font-face {
    font-family:      "Aurelis ADF Script No2 Std Condensed";
    font-weight:      normal;
    font-style:       italic;
    font-variant:     normal;
    src: local("Aurelis ADF Script No2 Std Cd It"),
        url("../fonts/opentype/AurelisADFScriptNo2Std-CdIta.otf");
}
@font-face {
    font-family:      "Aurelis ADF Script No2 Std Expanded";
    font-weight:      normal;
    font-style:       italic;
    font-variant:     normal;
    src: local("Aurelis ADF Script No2 Ext"),
        url("../fonts/opentype/AurelisADFScriptNo2Std-ExtIt.otf");
}
@font-face {
    font-family:      "Aurelis ADF Script No2 Std";
    font-weight:      normal;
    font-style:       italic;
    font-variant:     normal;
    src: local("Aurelis ADF Script No2 Std Italic"),
        url("../fonts/opentype/AurelisADFScriptNo2Std-Italic.otf");
}
...
<p class="style001">
    Les polices de caractères de la famille « Aurelis
```



```

ADF Script N&deg;2 Std&nbsp;&raquo; existent en version <span
class="style001"><i>Aurelis
ADF Script N&deg;2 Std normale;</></span>, <span
class="style002"><i>Aurelis
ADF Script N&deg;2 condens&eacute;</></span> et <span
class="style003"><i>Aurelis
ADF Script N&deg;2 Std expans&eacute;</></span>.</p>

```

Ce que le dispositif de lecture affichera :

Les polices de caractères de la famille « Aurelis ADF Script N°2 Std » existent en version *Aurelis ADF Script N°2 normale*, *Aurelis ADF Script N°2 condensée* et *Aurelis ADF Script N°2 expansée*.

Ajustement « automatique » des tailles de polices de caractère

Les deux polices de caractères les plus utilisées dans les paragraphes des autres chapitres ce document sont les polices de caractères « Alfios » et « Verily Serif Mono », de style « roman »—ou « normal » . Les valeurs du rapport hauteur de minuscules/hauteur de majuscules de ces deux polices de caractères sont respectivement 0.37 et 0.52. L'ajustement de la taille de la police de caractères « Verily Serif Mono » à celle de la police de caractères « Alfios » donne la taille de 12pt*0.37/0.52 soit approximativement 8.54pt ou encore 8.54pt/12pt soit environ 0.712em, pour la première de ces polices de caractères. Dans la version de ce « Guide de l'Utilisateur » que vous êtes en train de lire, l'ajustement des tailles de polices de caractères entre elles est effectué, pour la totalité du document, avec l'option « -x » de la ligne de commande (voir chapitre suivant, « »).

Le code XHTML et CSS 3.0 ci-dessous montre comment indiquer à whatever d'avoir à calculer et appliquer ponctuellement l'ajustement de la taille de la police de caractères « Verily Serif Mono » à celle de la police de caractères « Alfios », quand l'option d'ajustement automatique n'est pas utilisée :

```

p {
  font-family:      serif, sans-serif, monospace;
  font-size:        12pt;
  font-size-adjust: 0.37;
}
code.style001 {
  font-family:      monospace, serif, sans-serif;
}
...
<p>
  Un exemple d'utilisation de <code
  class="style001">font-size-adjust</code>.</p>

```

Qui sera traduit en :

```

p {
  font-family:      serif, sans-serif, monospace;
  font-size:        12pt;
}
span.style001 {
  font-family:      monospace, serif, sans-serif;
  font-size:        0.712em;
}

```

...

<p>

Un exemple d'utilisation de <span
class="style001">font-size-adjust.</p>

Et affiché :

Un exemple d'utilisation de font-size-adjust.

Create documents with the *Writer* word processor from the *OpenOffice* suite, save them as HTML et and convert them to “standard” ePUB et PDF electronic books is what the “whatever” tool makes available to you. Its main features include:

- 🐼 ePUB and/or PDF electronic books composition from the same source files. Structure and content are—a few exceptions not withstanding—identical, while the printable version has enhancements, which bring it closer to hand-made books.
- 🐼 A shared ePUB file's CSS stylesheet true to the source styles, as all properties and attributes « whatever » understands are taken into account, while it allows for scaling, as all dimensions are relatives either to the page size, the body font, or the current font, as needs be.
- 🐼 Quasi-professional quality PDF documents, thanks to the T_EX typographic macro-language—widely acknowledged as one of the most sophisticated ones—on which ConT_EXt relies. ConT_EXt MkIV version is the only software of its kind which uses *OpenType* et *TrueType* fonts out-of-the-box—as does *OpenOffice Writer*—and provides microtypographic features.
- 🐼 An “expert” mode, which gives access to all weights and widths available within a font family, provides for font size's ajustement between heterogeneous fonts, for better reading, while the ePUB produced file still follows the CSS 2.1 specifications, who do not provide such capabilities.

It also aims at as much exhaustivity as the constraint of ePUB and PDF similiarity permits. Its current version builds electronic books with:

- 🐼 Up to six title levels.
- 🐼 A customizable table of contents.
- 🐼 An index (PDF documents).
- 🐼 A bibliography (PDF documents).
- 🐼 Headers and footers (PDF documents).
- 🐼 “Simple” tables.
- 🐼 Definitions.
- 🐼 Up to four levels of ordered or unordered lists.
- 🐼 Floated and fixed graphics.
- 🐼 Footnotes.
- 🐼 References.
- 🐼 Horizontal rules.
- 🐼 « verbatim » paragraphs.
- 🐼 Multi-lingual text, with appropriate typography and hyphenation.
- 🐼 Rich text including *italics*, **bold**, underscores, ~~strikes~~, ^{superscripts}, and _{subscripts}.
- 🐼 Customized bullets (PDF documents).

Contents

The following chapters are a step-by-step tutorial on writing documents to be composed by “whatever” with *OpenOffice Writer* and « expert » mode's use of XHTML and CSS files “whatever” may produces as well. The sources files for the present guide, which its distribution includes, can also be looked at as examples and used to test your software « ».

- 🐼 Chapter , « » explains how to install the software and related tools, and then rebuild this guide to test your installation.
- 🐼 Chapter , « » introduces the few notions you must know before writing your first documents.

- 🔗 Chapter , « » deals with formatting the most usual elements of your documents using *OpenOffice Writer*.
- 🔗 Chapter , « » treats of more complex elements, such as tables, references, etc. If you plan to use *OpenOffice Writer*, exclusively, you then can skip the next chapter.
- 🔗 Chapter is dedicated to the « », which was but evoked until there. Appendice , « », sums up HTML to XHTML conversion with a few tables. You should consult them only if you wish to work this way.
- 🔗 Chapter , « » describes parameters and specificities of PDF documents composition.
- 🔗 Chapter , « » details *CSS Font Module Level 3* compatibility, when using “expert” mode.
- 🔗 Chapter , « » lists command line options and their effects.

Remerciements

ConT_EXt est développé depuis 1990 par Hans Hagen de PRAGMA Advanced Document Engineering (Pragma ADE), une société des Pays-Bas.

Lua a été créé en 1993 par Roberto Ierusalimsky, Luiz Henrique de Figueiredo, et Waldemar Celes, membres du groupe de recherche Tecgraf, de l'Université Catholique Pontificale de Rio de Janeiro, au Brésil.

Les familles de polices de caractères « Alfios », « Akkadian », et « Gardiner », appartiennent aux « Polices de Caractères pour les Écritures Archaiques » de George Douros.

Les familles de polices de caractères « Aurelis ADF N°2 », « Aurelis ADF Script N°2 », « Ornaments ADF », et « Keypad ADF », ont été créés par Hirwen Harendal, d'Arkandis Digital Foundry.

La police de caractères « Verily Serif Mono » est l'oeuvre de Stephen Hartke, de l'université de Lincoln, Nebraska.

Acknowledgements

ConT_EXt was developed from 1990 by Hans Hagen from PRAGMA Advanced Document Engineering (Pragma ADE), a Netherlands-based company.

Lua was created in 1993 by Roberto Ierusalimschy, Luiz Henrique de Figueiredo, and Waldemar Celes, members of the Computer Graphics Technology Group (Tecgraf) at the Pontifical Catholic University of Rio de Janeiro, in Brazil.

The “Alfios”, “Akkadian”, and “Gardiner” font families are part of George Douros’s “Unicode Fonts for Ancient Scripts”.

The “Aurelis ADF N°2”, “Aurelis ADF Script N°2”, “Ornaments ADF”, and “Keypad ADF” have been designed by Hirwen Harendal of Arkandis Digital Foundry.

The “Verily Serif Mono” font was created by Stephen Hartke, from the University of Lincoln, Nebraska.

- C
 - Context 19
 - MkIV 19
- H
 - HTML
 - entités 14
- I
 - ISO-8859-1 14
- M
 - micro-typography 19
 - mode
 - expert 19
- P
 - police(s) de caractères
- catalogue 7
- chasse 8
 - modification 11
- famille
 - générique
 - de base 7
- sélection 9
- taille
 - ajustement 13
- polices(s) de caractères
 - sélection
 - approchée 12
- ponctuation
 - caractères 10
- S
 - scaling 19
- T
 - Tex 19
 - texte
 - mot 9
- U
 - Unicode
 - paire de substitutions 14
 - point de code 14
 - UTF-8 14
- font-face 8

Bibliographie

- ¹ Hans Hagen. *ConT_EXt, the manual*. Pragme ADE, Hasselt , 2001.
- ² Taco Hoekwater *meeting*. *ConT_EXtGarden.net/2008/talks/2008-08-21-taco-bib/bibtalk.pdf*.
- ³ Taco Hoekwater. Bibliographies. Dans *ConT_EXt, Module Documentation*. Pragme ADE, Hasselt, 2009.
- ⁴ Ton Otten et Hans Hagen. *ConT_EXt, an excursion*. Pragme ADE, Hasselt , 1997.
- ⁵ Norman Walsh Help on bibT_EX. *<http://nwalsh.com/tex/texhelp/BibT_EX.html>*.

