New Features in mPDF v6.0

Advanced Typography

Many TrueType fonts contain OpenType Layout (OTL) tables. These Advanced Typographic tables contain additional information that extend the capabilities of the fonts to support high-quality international typography:

- OTL fonts support ligatures, positional forms, alternates, and other substitutions.
- OTL fonts include information to support features for two-dimensional positioning and glyph attachment.
- OTL fonts contain explicit script and language information, so a text-processing application can adjust its behavior accordingly.

mPDF 6 introduces the power and flexibility of the OpenType Layout font model into PDF. mPDF 6 supports GSUB, GPOS and GDEF tables for now. mPDF 6 does not support BASE and JSTF at present.

Other mPDF 6 features to enhance complex scripts:

- improved Bidirectional (Bidi) algorithm for right-to-left (RTL) text
- support for Kashida for justification of arabic scripts
- partial support for CSS3 optional font features e.g. font-feature-settings, font-variant
- improved "autofont" capability to select fonts automatically for any script
- support for CSS :lang selector
- dictionary-based line-breaking for Lao, Thai and Khmer (U+200B is also supported)
- separate algorithm for Tibetan line-breaking

Note: There are other smart-font technologies around to deal with complex scripts, namely Graphite fonts (SIL International) and Apple Advanced Typography (AAT by Apple/Mac). mPDF 6 does not support these.

What can OTL Fonts do?

Support for OTL fonts allows the faithful display of almost all complex scripts:

- Arabic (السلام عليكم), Hebrew (שלום), Syriac (ܫܠܡ ܥܠܝܟܘܢ)
- Indic - Bengali (স্লামালিকুম), Devanagari (नमस्ते), Gujarati (નમસ્તે), Punjabi (ਸਤਿ ਸ੍ਰੀ ਅਕਾਲ), Kannada (ನಮಸ್ತೆ), Malayalam (നമസ്ക്കാരം), Oriya (ନମସ୍କାର), Tamil (வணக்கம்), Telugu (నమస్కారం)

- Sinhala (අඉඌඔබඝ), Thai (สวัสดี), Lao (ສະບາຍດີ), Khmer (ជំរាបសួរ), Myanmar (မင်္ဂလာပမ္ဍ), Tibetan (བཀྲ་ཤིས་བདེ་ལེགས།)

Joining and Reordering

র + ◌্ + খ + ◌্ + ম + ◌্ + ক + ◌্ + ষ + ◌্ + র + ি + ু = খ্মিষ্ক্রু

cf. http://www.microsoft.com/typography/OpenTypeDev/bengali/intro.htm

Ligatures

ffi ffl fi ffi ffl fi

Language-dependent substitutions

Arabic: ۴ ۶ ۷ Urdu: ۴ ٦ ۷ Arabic: ه ﻭ ﭘ ه Kurdish: ھ ﺋﻮ ﺋﭙ ﺋه

Font features - Optional substitutions

Stylistic Alternatives (salt)

Arabic: ۀ ھ ہ ک ع ئ Farsi: ۀ ہ ہ ک ع ے Arabic: گ Turkish: ك

CSS control of discretionary OTL features

salt: (off) all (on) all

frac: (off) 1/4 3/10 (on) ¼ ³⁄₁₀

zero: (off) 1,000 (on) 1,000

onum: (off) 0123456789 (on) 0123456789

sups: (off) (32) (on) $^{(32)}$

Stylistic Alternatives (ss03,ss04): (off) अ झ ण झ ९ (on) �short झ ण झ ९

A full list of feature tags is at http://www.microsoft.com/typography/otspec/featurelist.htm

In mPDF, the following features are on by default:

- GSUB features: locl ccmp pref blwf abvf pstf pres abvs blws psts haln rlig calt liga clig mset (all scripts)
- GSUB features: isol fina fin2 fin3 medi med2 init nukt akhn rphf rkrf half vatu cjct cfar (for appropriate scripts e.g. Indic, Arabic)
- GPOS features: abvm blwm mark mkmk curs cpsp dist requ [kern]

NB 'requ' is not listed in the Microsoft registry of Feature tags; however it is found in the Arial Unicode MS font (it repositions the baseline for punctuation in Kannada script).

Kern is used in some fonts to reposition marks etc. and is essential for correct display, so in mPDF kern is on by default when any non-Latin script is used.

Mark repositioning (and diacritics)

ٱ ٱ ٱ ٱ

Mark repositioning (and Contextual substitution)

Ǻ á ï Á á ï

Complex syllables

Note that the text displayed is dependent on the font's design/capabilities. These are both "correct" representations of the same string, using:

FreeSerif: र्द्मि and FreeSans font: द्र्मि

cf. http://www.microsoft.com/typography/OpenTypeDev/devanagari/intro.htm

Complex Typography

An example which utilises many different GSUB and GPOS features together - first without GSUB and GPOS:

تسـمـوا أو كبيرًا إلى لله وأدنى ألا إلآ بينكم عليكم ألا يعتم

With GSUB and GPOS:

تَسْـمُوٓا أَوْ كَبِيرًا إِلَىٰٓ للَّهِ وَأَدْنَىٰٓ أَلَّا إِلَّآ بَيْنَكُمْ عَلَيْكُمْ أَلَّا يَعْتُمْ

Text Justification using Kashida

يَـٰٓأَيُّهَا لَّذِينَ ءَامَنُوٓا إِذَا تَدَايَنتُم بِدَيْنٍ إِلَىٰٓ أَجَلٍ مُّسَمًّى فَكْتُبُوهُ

يَـٰٓأَيُّهَا لَّذِينَ ءَامَنُوٓا إِذَا تَدَايَنتُم بِدَيْنٍ إِلَىٰٓ أَجَلٍ مُّسَمًّى فَكْتُبُوهُ

What is "correct"?

There are a number of factors which determine how the input text is displayed in an application.

The font's capabilities/design (this example shows the same text input shown in 2 fonts):

FreeSerif: द्‌मि and FreeSans font: द्‌र्मि

Complex scripts require a "shaping engine" to re-order glyphs and apply the OTL features by syllable. MS Word and Wordpad run on the Windows platform use "Uniscribe", whereas some browsers such as FireFox and OpenOffice use Pango/HarfBuzz. The different shaping engines (and indeed different versions of them) can produce different results.

Different applications have different defaults (on/off) for some of the features e.g. kerning.

When testing mPDF 6, if text does not appear as you expect, ensure that the font is installed on your computer, and view the HTML in a browser. Also try copying/pasting the text into Wordpad/Word/OpenOffice and ensure that the correct font has been applied.

Note that Wordpad sometimes substitutes a different font if it does not like the one you have chosen, and does not even indicate that the substitution has occurred.

CSS control of font features

See http://www.w3.org/TR/css3-fonts/#font-rend-props for information about CSS3 and font-features.

The following are supported in mPDF 6:

- font-variant-position
- font-variant-caps
- font-variant-ligatures
- font-variant-numeric
- font-variant-alternates - Only [normal | historical-forms] supported (i.e. most are NOT supported) e.g. stylistic, styleset, character-variant, swash, ornaments, annotation (use font-feature-settings for these)
- font-variant - as above, and except for: east-asian-variant-values, east-asian-width-values, ruby
- font-language-override
- font-feature-settings

font-variant-east-asian is NOT supported

NB @font-face is NOT supported

NB @font-feature-values is NOT supported

Note that font-variant specifies a single property in CSS2, whereas in CSS3 it has become a shorthand for all the other font-variant-* properties. `font-variant: small-caps` was the only one supported in mPDF <v6, and will still work in mPDF 6.

See notes later about font kerning.

Some examples

```
/* use small-cap alternate glyphs */
.smallcaps { font-feature-settings: "smcp" on; }

/* convert both upper and lowercase to small caps (affects punctuation also) */
.allsmallcaps { font-feature-settings: "c2sc", "smcp"; }

/* enable historical forms */
.hist { font-feature-settings: "hist"; }

/* disable common ligatures, usually on by default */
.noligs { font-feature-settings: "liga" 0; }

/* enable tabular (monospaced) figures */
td.tabular { font-feature-settings: "tnum"; }

/* enable automatic fractions */
.fractions { font-feature-settings: "frac"; }

/* use the second available swash character */
.swash { font-feature-settings: "swsh" 2; }

/* enable stylistic set 7 */
.fancystyle {
font-family: Gabriola; /* available on Windows 7, and on Mac OS */
font-feature-settings: "ss07";
}
```

More Examples

Note the automatic line breaking used in Lao, Thai, Khmer and Tibetan text.

SYRIAC - Estrangelo Edessa

ܟܬܒ ܟܬܝܢ ܠܟ݂ܠܗ݂ ,ܐܝܟ ܟܬܠܐ ܟܬܒܐ ܒܬܒ ܟܬܒܐ ܟܝܢ ܟܐ ܒܫܒܫܐ ܕܗܒܐ
ܟܬܒܐ (ܟܬܫܒܝܐ ܟܬܫܒܐ ܗܢ) ܟܝܢ ܠܒܢ ܟܬܒ ܒܬ ܠ . ܐܝܟ ܒ
ܒܬܒ ; ܟܝܒܬܒ ܠܒܐ ܟܠܢ ; ܟܝܠܐ ܠܟ݂ܠܗ݂ ,ܐܝܟ ܟܝܢ ܒ ܟܒܐܝ ܟܬܝ
ܒܬ ,ܐܝܟ ܒܬ ܟܬܝܠܐ ܗ݂ ܟܝܒܐ ; ܟܝܢ ܒܫܒܫܐ ܕܗܒܐ .ܐܒܬܐ ; ܟܬܠܫܐ ܠ
.ܐ 473 ܒܫܒ ܟܝܢܠ ܐܟܒܫܐ ܟܝܢ (ܠܬܫܐܠ)

MYANMAR (Burmese) Padauk Book (SIL Font)

မြန်မာအခေါ် တရားဝင်အားဖြင့် ပြည်ထောင်စု သမ္မတ မြန်မာနိုင်ငံတော်သည် အရှေ့တောင်အာရှတွင် ဧရိယာအား ဖြင့် ဒုတိယ အကျယ်ဝန်းဆုံး[၁] တိုင်းပြည် ဖြစ်သည်။ ၁၉၄၈ ခုနှစ် ဇန်နဝါရီ ၄ ရက်တွင် ဂရိတ်ဗြိတိန်နိုင်ငံထံမှ (အင်္ဂလိပ်လို "Myanmar" အဖြစ်နှင့်) ပြည်ထောင်စုမြန်မာနိုင်ငံတော်အဖြစ် လွတ်လပ်ရေးကို ရရှိခဲ့သည်။ နောက်ပိုင်းတွင် ပြည်ထောင်စု ဆိုရှယ်လစ် သမ္မတမြန်မာနိုင်ငံတော်အဖြစ် ၁၉၇၄ ခုနှစ် ဇန်နဝါရီ ၄ ရက်တွင်လည်းကောင်း၊ ပြည်ထောင်စု မြန်မာနိုင်ငံတော်အဖြစ် ၁၉၈၈ခုနှစ် စက်တင်ဘာ ၂၃ ရက်တွင်လည်းကောင်း၊ ပြည်ထောင်စု မြန်မာနိုင်ငံတော်အဖြစ် ၁၉၈၉ ခုနှစ် ဇွန် ၁၈ ရက်တွင် လည်းကောင်း အမည်များပြောင်းလဲခဲ့သည်။ အာဏာရစစ်အစိုးရအား အသိအမှတ် မပြုသော အဖွဲ့အစည်းများက ဘားမား ("Burma") ဟုသာ အသိအမှတ်ပြု သုံးစွဲခဲ့သည်။ နိုင်ငံတော်အလံကိုလည်း ယခင် နိုင်ငံတော် အေးချမ်းသာယာရေးနှင့် ဖွံ့ဖြိုးရေးကောင်စီအစိုးရလက်ထက် ၂၀၀၈ ခုနှစ် ဖွဲ့စည်းပုံ အခြေခံ ဥပဒေတွင် ပြဋ္ဌာန်းထားသည့် ပြည်ထောင်စု သမ္မတ မြန်မာနိုင်ငံတော်အလံ ဖြင့် ၂၀၁၀ ခုနှစ် အောက်တိုဘာလ ၂၁ ရက်နေ့တွင် အလံစတင်လွှင့်ထူခြင်း အခမ်းအနားများကို နိုင်ငံတဝှမ်း ကျင်းပကာ အစားထိုး၍ ပြောင်းလဲ အသုံးခဲ့ပြုသည်။

KHMER

ង្គ្រៃ ន្ត្រី រង្គោះ

យុវជនម្នាក់បានស្លាប់ដោយគ្រាប់កាំភ្លើងរបស់ប៉ូលិស និងបីនាក់ផ្សេងទៀតរងរបួស នៅក្នុងការប្រឈមមុខដាក់គ្នាដាច់ដោយឡែកគ្នាមួយ រវាងក្រុមយុវជនមួយក្រុម ជាមួយប៉ូលិស នៅម្តុំស្ពានក្បាលថ្ពល់។ នេះបើតាមព័ត៌មានពីលោក ចាន់ សាវ៉ែត មន្ត្រីស៊ើបអង្កេតរបស់អង្គការសិទ្ធិមនុស្សអាដហុក ដែលវត្តមាននៅកន្លែងកន្លែងកើតហេតុ នៅយប់ថ្ងៃអាទិត្យទី ១៥ កញ្ញានេះ។

HEBREW - with Niqud and T'amim (cantillation)

לָכֵן חַכּוּ־לִי נְאֻם־יְהוָה לְיוֹם קוּמִי לְעֵד כִּי מִשְׁפָּטִי לֶאֱסֹף גּוֹיִם לְקָבְצִי מַמְלָכוֹת לִשְׁפֹּךְ עֲלֵיהֶם
זַעְמִי כָּל חֲרוֹן אַפִּי כִּי בְּאֵשׁ קִנְאָתִי תֵּאָכֵל כָּל־הָאָרֶץ׃

NKo

ߒ ߓ ߞ ߊ ߟ ߋ ߘ ߏ ߟ ߋ ߘ ߏ ߟ ߋ ߘ ߏ

ᠬᠡᠷᠪᠡ ᠪᠢᠳᠡ ᠮᠠᠨᠤ ᠬᠡᠯᠡᠨ ᠳᠡᠭᠡᠷ ᠬᠡᠯᠡᠭᠰᠡᠨ ᠦᠭᠡᠨ ᠪᠦᠷ ᠨᠢ ᠪᠢᠳᠡᠨᠢ ᠭᠡᠵᠦ ᠪᠣᠳᠣᠵᠤ ᠪᠠᠢᠨ᠎ᠠ ᠃ ᠬᠡᠷᠪᠡ ᠪᠢᠳᠡ ᠮᠠᠨᠤ ᠬᠡᠯᠡᠨ ᠳᠡᠭᠡᠷ ᠬᠡᠯᠡᠭᠰᠡᠨ ᠦᠭᠡᠨ ᠪᠦᠷ ᠨᠢ ᠪᠢᠳᠡᠨᠢ ᠭᠡᠵᠦ ᠪᠣᠳᠣᠵᠤ ᠪᠠᠢᠨ᠎ᠠ ᠃

LAO

ສາທາລະນະລັດ ປະຊາທິປະໄຕ ປະຊາຊົນ ລາວ (ຄໍາເຄົ້າ: ສາ ທາ ຣນ ຣັຖ ປ ຊາ ທິ ປ ໄຕ ຍ ປ ຊາ ຊົນ ລ າ ວ [໑]) ຫຼື ສປປ ລາວ ຕັ້ງ ຢູ່ ທິດ ຕາ ເວັນ ອອກ ສ ຽ ງ ໃຕ້ ຂ ອ ງ ທ ະ ວີ ບ ອ າ ຊີ , ຢ ູ່ ໃ ຈ ກ າ ງ ຂ ອ ງ ແ ຫ ຼ ມ ອິ ນ ດ ູ ຈ ີ ນ ລ ະ ຫ ວ ່ າ ງ ເ ສ ັ ້ ນ ຂ ະ ໜ ານ ທ ີ 14 - 23 ອ ົ ງ ສ າ ເ ໜ ື ອ ແ ລ ະ ເ ສ ັ ້ ນ ແ ອ ວ ທ ີ 100-108 ອ ົ ງ ສ າ ສ ປ ປ ລ າ ວ ມ ີ ເ ນ ື ້ ອ ທ ີ ່ ທ ັ ງ ໝ ົ ດ 236. 800 ຕ າ ລ າ ງ ກ ິ ໂ ລ ແ ມ ັ ດ ເ ປ ັ ນ ປ ະ ເ ທ ດ ທ ີ ່ ບ ໍ ່ ມ ີ ທ າ ງ ອ ອ ກ ສ ู ่ ທ ะ ເ ล , ม ี ช า ย แ ด น ต ิ ด ก ั บ ส า ท า ล ะ น ะ ล ั ด ป ะ ช า ช ื ่ น จ ี น (505 ก ิ โ ล แ ม ั ด), ท ิ ด ใ ต ้ ต ิ ด ก ั บ ล า ช ะ อ า ณ า จ ั ก ก ำ ป ู เ จ ย (435 ก ิ โ ล แ ม ั ด), ท ิ ด ต า เ อ ั น อ อ ก ต ิ ด ก ั บ ส า ท า ล ะ น ะ ล ั ด ส ั ง ค ม น ิ ย ม ว ี ย ด น า ม (2. 069 ก ิ โ ล แ ม ั ด), ท ิ ด ต า เ อ ั น ต ก ต ิ ด ก ั บ ล า ช ะ อ า ณ า จ ั ก ไ ท (1. 385 ก ิ โ ล แ ม ั ด), แ ล ะ ท ิ ด ต า เ อ ั น ต ก ส ฽ ง เ ห ื อ ต ิ ด ก ั บ ส า ท า ล ะ น ะ ล ั ด ແ ຫ ່ ງ ส ะ ห ะ พ า บ ม ฽ น ม า (236 ก ิ โ ล แ ม ั ด), ส. ป. ป. ล า ว ເ ป ็ น ป ะ ເ ท ด ด ฽ อ ใ น พ า ก ฽ พ ื ้ น ม ี ้ ท ี ่ บ ่ ม ี ช า ย แ ด น ต ิ ด ก ั บ ท ะ ล ะ .

THAI

"ซัมติง" เป็นเพลงของวงเดอะบีตเทิลส์ ในปี ค.ศ. 1969 เป็นเพลงที่บรรจุอยู่ในอัลบั้มชุด แอบบีโรด เพลงนี้เป็นเพลงแรกที่ซิงเกิลหน้าเอที่จอร์จ แฮร์ริสันเขียน และถือเป็นซิงเกิลแรกของเดอะบีตเทิลส์ที่มีเพลงที่มีอยู่แล้วในอัลบั้มบรรจุอยู่ด้วย ทั้งเพลง "ซัมติง" และเพลง "คัมทูเกตเตอร์" ที่อยู่ในอัลบั้ม แอบบีโรด และเพลง "ซัมติง" ถือเป็นเพลงเดียวที่แฮร์ริสันแต่งแล้วขึ้นอันดับ 1 บนชาร์ตอเมริกันขณะที่ยังอยู่ในวงเดอะบีตเทิลส์

SINHALA

නඩත්තු කාර්යයන් විකිපීඩියාව ප්රශස්ත මට්ටමකින් පවත්වා ගැනීම සඳහා අත්යවශ්ය අංගයකි. උපදෙස් රෑරාව මගින් තවත් අලුත් නඩත්තු කාර්යයන් තැනීමේදී පරිස්සම් විය යුතුය. ඔබට හැකි විට සංවිධානාත්මක වන්න, එහෙත් සෑමවිටම විශාල පින්තූරය ඇතුලතින් තබා ගන්න: අප මෙතැනට පැමිණ සිටින්නේ විශ්වකෝෂයක් තැනීම සඳහායි. ඔබ සිතන්නේ නඩත්තු කිරීම්වලට වඩා අනෙකුත් පැතිවලින් විකිපීඩියාවට දායක වීමට නම්, කරුණාකර විකිපීඩියා:විකිපීඩියාවට දායක වීම බලන්න. ඉක්මනින් අවශ්ය සහ ඉතා වැදගත් වන කාර්යයන් විකිපීඩියා:අතපසු වූ වැඩ හිඳි සොයාගැනීමට අවස්ථාව ඇත.

TIBETAN

ཞི་ཀྲ་བླ་ཨོཾ་ཀ་ཀ་པ་ཀ་པ་རཾ་ཀ་ཀོ་ཏ་རྒྱ་ཀ། ཨོཾ་སུ་ནི་སུ་ནི་མ་ཏྲ་སུ་ནི་ཡེ་སྭ་ཏྲ་ཨོཾ་ཁ་ཧི་མ་ཝ་ནྟ་ཀྱི་ཨོཾ་མ་ཀི་པ་ལྭ་ཏྲ་ཨོཾ་ཁ་ཧི། བཛྲ་ཀུ་ས་ལཱ་ཏྲ་ཀྱི་ཨོཾ་ཁ་ཧི་ཁ་ཧི་ཡ་ཏྲ་ཨོཾ་ཁ་ཧི་ཁ་ཧི་ཡ་ཏྲ་ཨོཾ་ཁ་ཧི་ཁ་ཧི་ཡ་ཏྲ་ཨོཾ་ཁ་ཧི་ཁ་ཧི་ཡ་ཏྲ་ཨོཾ་ཁ་ཧི། སྭ་ཏྲ་ཨོཾ་ཁ་ཧི་ཁ་ཧི་ཡ་ཏྲ་ཨོཾ་མ་རི་ཏྲ་ཨོཾ་ཁ་ཧི།

TAI THAM

ᩉᩣᩫᨦᩖᩫᩋᩬᨾᩣᨩᩥᨶᨯᩣ ᨦᨯᩣᨸᩩᩴᨦᩖᩫᨩᩥᨠᩫᩁᩖᩫᩋᩬᨾᩣᩃᩫ ᨲᩫᩉᩮᩣᨯᩖᩥᨸᩥᩉᩣᨾᩣᨦᨦᩮᩣ
ᨾᩣᨩᩫᨾᩣᨦᨶᩮᩣ ᨸᩫᩁᩫᨠᩫᩃᩫᨠᩪᨿ ᩉᩣᩫᨦᩖᩫᨦᩖᩫᩋᩫᩃᩥᨠᨠᩣᩫᨩᩥᨿ ᨦᩣᩫᨸᩩᩉᩣᨿᨻᩩᨶᩮᩣᨦᨠᩫᨾᩣᩃᩪᨿ
ᨦᩣᩫᨠᩣᨶᨯᨠᩫᨦᨦᩮᩣᩅᩩᨦᩋᩩᨯ ᨦᩣᩫᨠᩣᩁᩣᨦᩣᩫᨾᩪᨿ ᨦᩣᩫᨻᩩᨯᨦᩣᩫᨾᩪᨿ ᨦᩣᩫᨻᩩᨯᨲᩮᩣᩉᩣᨿᨾᩣᩃᩪᨿ
ᨦᩣᩫᨠᩮᩣᨶᩫᨦᨦᩮᩣᨾᩣᨦᨦᩮᩣᨾᩣᨦᨦᩮᩣᨾᩣ

Dictionary Line breaking

Lao, Thai and Khmer text does not have space between words. By default, mPDF 6 uses word dictionaries to determine appropriate opportunites for line-breaks. Users may turn this function off using the configurable variable `useDictionaryLBR`.

Alternatively users can insert the character U+200B (zero-width space) in the text to mark line-breaking opportunities manually.

Similarly for Tibetan script, mPDF 6 uses a simple algorithm to identify line-breaking opportunities after the characters U+0F0B (Tsheg) or U+0F0D. This can be overriden using the configurable variable `useTibetanLBR`.

Myanmar Fonts

Myanmar (Burmese) on the web is quite frequently written for fonts which are not strictly unicode-compliant. This includes common applications such as WordPress and a number of official Burmese government websites.

Ayar fonts (http://www.ayarunicodegroup.org) are based on text input where the vowel preceeds the consonant (which is contrary to Unicode specification).

ZawGyi-One is another very common font in use. This font has some characters incorrectly coded e.g. U+103A as U+1039.

There are also fonts available which are fully unicode compliant, such as Padauk, Tharlon, Myanmar3, and Microsoft's Myanmar Text.

As long as you select the right font for the input text, all of them work fine in mPDF:

> Tharlon: ဒီရက်ပိုင်းမှာ ဧရာဖောင့်ကို ယူနီကုဒ်အဖြစ် ရည်ညွှန်းပြောဆိုနေကြတာ တွေ့ရလို့ ဧရာဟာ ယူနီကုဒ် မဖြစ်ကြောင်းနဲ့ ဘာလို့မဖြစ်ရတာလဲဆိုတာ အတိုပဲ ရှင်းပါမယ်။ ယူနီကုဒ်ဖြစ်ဖို့ - ၁။ ယူနီကုဒ် ကုဒ်ပွိုင့်နဲ့ ကိုက်ညီရပါမယ်။ ၂။ ယူနီကုဒ် စာလုံးစီပုံ (Encoding) နဲ့ ကိုက်ညီရပါမယ်။

from http://www.myanmarlanguage.org/unicode

> Zawgyi-one: စီးပွားရေးနှင့်ကူးသန်းရောင်းဝယ်ရေးဝန်ကြီးဌာန ပြည်ထောင်စုဝန်ကြီး မန္တလေးတိုင်းဒေသကြီး ကျေးလက်ဒေသ အသေးစား ကုန်ထုတ်လုပ်ငန်းများ ကြည့်ရှုအားပေး

from http://www.commerce.gov.mm/

> Ayar: WordPress တရားဝင် မြန်မာဘာသာ စာမျက်နှာမှ ကြိုဆိုပါတယ်။ ! ရာနှုန်းပြည့် ဘာသာပြန်ထားသည့် WordPress မြန်မာ ဘာသာပြန်မူကို ဗားရှင်း ၃.၁ ဖြင့် စတင် ဖြန့်ချိလိုက်ပြီးသည့်နောက် ဆက်လက်၍ အဆင့်မြှင့်တင်မှု ဗားရှင်းများကို အချိန်နှင့်တစ်ပြေးညီ

from https://mya.wordpress.org/

lang selector

mPDF 6 supports use of the lang selector in CSS. All of the following are supported:

- :lang(fr)
- p:lang(fr)
- span:lang("syr")
- [lang="fr"]
- [lang='fr']
- p[lang=fr]
- p[lang="zh-TW"]

Note: [lang=zh] will match lang="zh-TW" and lang="zh-HK"

Limitation: class selectors and attribute selectors should be of equal specificity in CSS specification e.g.

```
:lang(syr) { color: blue; }
.syriac { color: red; }
```

should be of equal specificity, and thus apply whichever comes later in the CSS stylesheet. mPDF 6 however gives :lang priority over .class

The use of the lang attribute and CSS selector is now the recommended method for handling multi-lingual documents in mPDF 6.

lang HTML attribute

The HTML lang attribute has a number of uses:

- when OTL tables are being used for a font, the language from the lang attribute is used to select which OTL features are applied;
- used in conjunction with CSS lang selector to allow CSS styles to be applied;
- can be used in conjunction with `autoLangToFont` and `autoScriptToLang` (see below)

IETF tags should be used for lang which comply with the following:

- a 2 or 3 letter Language code, followed optionally by
- a hyphen and a 4 letter Script code, and or
- a hyphen and a 2 letter Region code
- i.e. [xx|xxx]{-Xxxx}{-XX}
- mPDF deals with IETF tags as case insensitive

Automatic font selection

Note: This functionality of mPDF has changed considerably in mPDF v6 and is not backwards compatible.

mPDF 6 has two functions which can be used together or separately:

`autoScriptToLang` - marks up HTML text using the lang attribute, based on the Unicode script block in question, and configurable values in `config_script2lang.php`.

`autoLangToFont` - selects the font to use, based on the HTML lang attribute, using configurable values in `config_lang2font.php`.

For automatic font selection, ideally we would choose the font based on the language in use. However it is actually impossible to determine the language used from a string of HTML text. The Unicode script block can be ascertained, and sometimes this tells us the language e.g. Telugu. However, Cyrillic script is used for example in many different languages. So the best we can do is base it on the script used. However, mPDF 6 does this in two stages via the "lang" attribute, because this allows the options of using either of the stages alone or together:

```
<p>English руćский язык پښتو</p>
```

↓ **autoScriptToLang** (config_script2lang.php) ↓

```
<p>English <span lang="und-Cyrl">руćский язык</span>
<span lang="ps">پښتو</span></p>
```

↓ **autoLangToFont** (config_lang2fonts.php) ↓

```
Uses "lang" to select font, and to determine OTL features applied
```

autoScriptToLang

```
$mpdf->autoScriptToLang = true;
$mpdf->baseScript = 1;
$mpdf->autoVietnamese = true;
$mpdf->autoArabic = true;
```

`$mpdf->baseScript = 1;` tells mPDF which Script to ignore. It is set by default to "1" which is for Latin script. In this mode, all scripts *except* Latin script are marked up with "lang" attribute. To select other scripts as the base, see the file /classes/ucdn.php

Using autoScriptToLang, mPDF detects text runs based on Unicode script block; using the values in `config_script2lang.php` it then encloses the text run within a span tag with the appropriate language attribute. For many scripts, the language cannot be determined: see the example above which recognises Cyrillic script and marks it up using und-Cyrl, which is a valid IETF tag, coding for language="undetermined", script="Cyrillic".

Two optional refinements are added: Vietnamese text can often be recognised by the presence of certain characters which do not appear in other Latin script langauges, and similarly analysis of the text can attempt to distinguish Arabic, Farsi, Pashto, Urdu and Sindhi. If active, the text will then be marked with a specific language tag e.g. "vi", "pa", "ur", "fa" etc.

These features can be disabled or enabled (default) using the variables `$mpdf->autoVietnamese` `$mpdf->autoArabic`, either in config.php or at runtime.

autoLangToFont

```
$mpdf->autoLangToFont = true;
```

You can edit the values in `config_lang2font.php` to specify which fonts are used for which "lang".

Using text with multiple languages

Recommended ways to use multiple languages in mPDF:

1. If you have full control over the HTML, mark-up the text with the "lang" atribute and use CSS (:lang selector preferably); this method means that the language information can also be used by OTL for language dependent substitutions.
2. If you have no control over (user) HTML input and want to output faithfully, use both autoScriptToLang and autoLangToFont

It is preferable not to use autoScriptToLang and autoLangToFont unless they are necessary: they will result in increased processing time, and OTL tables will not be able to use language dependent substitutions when undefined languages are set e.g "und-Cyrl".

Updating from previous mPDF versions

As a brief summary, to update from previous versions of mPDF:
Use $this->autoScriptToLang=true instead of $this->SetAutoFont()
Use $this->autoLangToFont instead of $this->useLang

Bidi Bidirectional text

The algorithm to handle bi-directional text (right to left) has been completely rewritten. Text is now processed across the whole paragraph ignoring inline tags. There is also full support for the methods to control/override the display.

1) The following Unicode characters are supported, and can be inserted directly in the text as HTML entities:

LRE	U+202A	LEFT-TO-RIGHT EMBEDDING	‪
RLE	U+202B	RIGHT-TO-LEFT EMBEDDING	‫
LRO	U+202D	LEFT-TO-RIGHT OVERRIDE	‭
RLO	U+202E	RIGHT-TO-LEFT OVERRIDE	‮
PDF	U+202C	POP DIRECTIONAL FORMATTING	‬
LRI	U+2066	LEFT-TO-RIGHT ISOLATE	⁦
RLI	U+2067	RIGHT-TO-LEFT ISOLATE	⁧
FSI	U+2068	FIRST STRONG ISOLATE	⁨
PDI	U+2069	POP DIRECTIONAL ISOLATE	⁩
LRM	U+200E	LEFT-TO-RIGHT MARK	‎
RLM	U+200F	RIGHT-TO-LEFT MARK	‏

2) The following HTML tags are supported:

- <bdo> (NB the "dir" attribute is mandatory on <bdo>)
- <bdi> (HTML5)

3) The CSS property "unicode-bidi" is supported with the following (CSS3) values: normal | embed |

isolate | bidi-override | isolate-override | plaintext.
See http://www.w3.org/TR/css3-writing-modes/#unicode-bidi for more details.
"unicode-bidi" is supported on block level elements as well as in-line elements, but note that:

- the value is not inherited to child blocks
- using "embed" or "isolate" has no effect on block level boxes
- "isolate-override" is equivalent to "bidi-override" on block level boxes

NB dir="auto" is not supported generally, but it is supported for <bdi> (has the same effect as if omitted) to use First Strong Isolate (FSI).

Directionality can now be set on individual table cells <td style="direction:rtl;unicode-bidi:embed;"> or <td dir="rtl">

Equivalent methods

The following are equivalent methods:

EMBED

```
<span dir="rtl">...</span>
&#x202B;...&#x202C;
<span style="direction: rtl; unicode-bidi: embed">...</span>
```

OVERRIDE

```
<bdo dir="rtl">...</bdo>
&#x202E;...&#x202C;
<span dir="rtl" style="unicode-bidi: bidi-override">...</span>
<span style="direction: rtl; unicode-bidi: bidi-override">...</span>
```

ISOLATE

```
<bdi dir="ltr">...</bdi>
&#x2067;...&#x2069;
<span dir="rtl" style="unicode-bidi: isolate">...</span>
<span style="direction: rtl; unicode-bidi: isolate">...</span>
```

First Strong Isolate (FSI)

```
<bdi>...</bdi>
<bdi dir="auto">...</bdi>
&#x2068;...&#x2069;
<span dir="rtl" style="unicode-bidi: plaintext">...</span>
<span style="direction: rtl; unicode-bidi: plaintext">...</span>
```

First strong isolate (FSI)

FSI is useful when including text within a paragraph where the directionality of the text is unknown. For example, if you are printing out a catalogue from a database of book titles and the number of readers, when some book titles are in right-to-left script, you may use this template:

```
<li>Title: {TITLE} - {READERS} readers</li>
```

This would result in the following:

- Title: Alice in Wonderland - 12390 readers
- Title: 17890 - עליסה בארץ הפלאות, סיפור-ילדים מאת לואיס קרול readers

```
<li>Title: <bdi>{TITLE}</bdi> - {READERS} readers</li>
```

Using BDI will result in the following:

- Title: Alice in Wonderland - 12390 readers
- Title: עליסה בארץ הפלאות, סיפור-ילדים מאת לואיס קרול - 17890 readers

Kerning

Kerning is a bit complicated! CSS3 allows for 2 methods of specifying kerning. In mPDF 6, these 2 methods have exactly the same effect:

- font-kerning: normal;
- font-feature-settings: 'kern' on;

TrueType fonts allow for 2 possible ways of including kerning data:

- OTL GPOS table may contain kerning information
- A separate kern table

Most fonts contain both or none, but they may exist independently.

If kerning is set to be active (by either of the CSS methods):

- if the useOTL value means that OTL GPOS tables are applied, then this method will be used;
- if not, then the separate kern table will be used - if it exists.

In Latin script, kerning will only be applied if specified by CSS. The configurable variable `useKerning` determines behaviour if `font-kerning: auto` is used (the default).

When using OTL tables, kerning is set to be on by default for non-LATIN script; this is because a number of fonts use information in the kern feature to reposition glyphs which are essential for correct display in complex scripts.

Limitation: if useOTL is set, but not for Latin script (e.g. = 0x02), and the text string contains more than one script, then kerning will not be applied to the Latin script text e.g. [Cyrillic text][Latin text][Cyrillic text]. This is because mPDF uses the presence of any repositioning applied to determine if kerning has been applied, otherwise using the alternative kern tables.

Small-Caps

Small Caps should be selected using:

```
<p style="font-variant-caps:small-caps">This is in small caps</p>
```

and will appear as: THIS IS IN SMALL CAPS

Note: `font-variant:small-caps` will also be recognised as font-variant is now considered the shorthand version cf. above.

If the font has useOTL enabled (to any value), and the font OTL tables contain the "smcp" feature, then the OTL feature will be used to substitute purpose-designed glyphs from the font. Otherwise, mPDF generates small capitals as in previous version.

Superscript and Subscript

```
<p>This is in <span style="font-variant-position:super">superscript</span></p>
```

will appear as superscript (only) if the font is OTL-capable and contains specific glyphs for superscript.

Note that font-variant:super will also be recognised as font-variant is now considered the shorthand version cf. above.

If the font has useOTL enabled (to any value), and the font OTL tables contain the "sups" feature, then the OTL feature will be used to substitute purpose-designed glyphs from the font.

The same for subscript using `font-variant-position:sub`.

If you wish to use a superscript/subscript which will work with any font, continue to use the tags <sup> and <sub> which (through the default CSS in config.php) will generate superscript using CSS vertical-align=super and font-size=55%.

How to use OTL in mPDF

In `config_fonts.php` there are 2 new variables which affect OTL features e.g.:

```
"dejavusanscondensed" => array(
'R' => "DejaVuSansCondensed.ttf",
'B' => "DejaVuSansCondensed-Bold.ttf",
'I' => "DejaVuSansCondensed-Oblique.ttf",
'BI' => "DejaVuSansCondensed-BoldOblique.ttf",
'useOTL' => 0xFF,
'useKashida' => 75,
),
```

mPDF is published with a large collection of fonts, and all configured to use their full OTL capabilities.

useOTL

useOTL should be set to an integer between 0 and 255. Each bit will enable OTL features for a different group of scripts:

Bit	dec	hex	Enabled
1	1	0x01	GSUB/GPOS - Latin script
2	2	0x02	GSUB/GPOS - Cyrillic script
3	4	0x04	GSUB/GPOS - Greek script
4	8	0x08	GSUB/GPOS - CJK scripts (excluding Hangul-Jamo)
5	16	0x10	(Reserved)
6	32	0x20	(Reserved)
7	64	0x40	(Reserved)
8	128	0x80	GSUB/GPOS - All other scripts (including all RTL scripts, complex scripts etc)

Setting useOTL to 0 (or omitting it) will disable all OTL features. Setting useOTL to 255 or 0xFF will enable OTL for all scripts. Setting useOTL to 0x82 will enable OTL features for Cyrillic and complex scripts.

In a font like Free Serif, it may be useful to enable OTL features for complex scripts, but disable OTL for Latin scripts (to save processing time). However, see above - this may disable kerning in Latin scripts in certain circumstances.

useKashida

useKashida should be set for arabic fonts if you wish to enable text justification using kashida. The value should be an integer between 0 and 100 and represents the percentage of additional space required to justify the text on a line as a ratio of kashida/inter-word spacing.

Choosing fonts to add to mPDF 6

Fonts with OTL need to have GDEF, GSUB and GPOS tables in the font file. Although TrueType font files are binary files, the table names and script/feature tags are written as ASCII characters; open the .ttf or .otf file in a text editor such as Windows Notepad, and you will see GDEF, GSUB and GPOS in the first few lines if they are present. You can also search the file to see if the script tags are present for your desired scripts cf. http://www.microsoft.com/typography/otspec/scripttags.htm.

Note: The OTL specification for Indic fonts was updated in 2005 to version 2. The v2 script tag for Bengali is "bng2" whereas prior to this it was "beng". Many open-source font files are still written for the old specification. This is supported by mPDF, although v2 fonts give better results.

Note: mPDF does not support Graphite or AAT font features.

Configuring new fonts for mPDF 6

To add a font, first copy the font file to the /ttfonts/ folder.

Then edit config_fonts.php to add. See the manual for details if you are not already familiar with this.

If you wish to use this font with autoLangToFont, you also need to edit config_lang2fonts.php

Setting OTL use at runtime

mPDF caches some font information in the /ttfontdata/ folder to improve performance. This is regenerated if you change the value of useOTL for a font.

There may be circumstances when you wish to use OTL features with different scripts depending on the document e.g. for everyday use you may want to disable OTL for FreeSerif to save processing time, but on occasions use OTL for Indic and/or Arabic scripts. The recommended way to do this is to create 2 instances of the font e.g. in config_fonts.php:

```
"freeserif" => array(
'R' => "FreeSerif.ttf",
'B' => "FreeSerifBold.ttf",
'I' => "FreeSerifItalic.ttf",
'BI' => "FreeSerifBoldItalic.ttf",
'useOTL' => 0x00,
),
"freeserif2" => array(
'R' => "FreeSerif.ttf",
'B' => "FreeSerifBold.ttf",
'I' => "FreeSerifItalic.ttf",
'BI' => "FreeSerifBoldItalic.ttf",
'useOTL' => 0xFF, /* Uses OTL for all scripts */
'useKashida' => 75,
),
```

You could then either use this second font name in your stylesheets e.g.

```
<p style="font-family:freeserif2;">Hallo World (in Arabic)</p>
```

or, you could use font translation e.g.

```
$mpdf->fonttrans['freeserif'] = 'freeserif2';
```

Page breaking

Types of page break

The handling of borders and padding at page breaks has been updated. mPDF has three types of page breaks:

1) "slice" - no border and no padding are inserted at a break. The effect is as though the element were rendered with no breaks present, and then sliced by the breaks afterward

2) "cloneall" - each page fragment is independently wrapped with the borders and padding of all open elements.

3) "clonebycss" - open elements which have the (custom) CSS property "box-decoration-break" set to "clone" are independently wrapped with their border and padding.

The difference between 2) and 3) is illustrated by this example:

```
<style>
div { border: 1px solid black; padding: 1em; }
.level1 { box-decoration-break: slice; }
.level2 { box-decoration-break: clone; }
.level3 { box-decoration-break: clone; }
</style>


<div class="level1">
<div class="level2">
<div class="level3">
<p style="page-break-after:always">...</p>
<p>....</p>
</div>
</div>
</div>
```

At the forced pagebreak which occurs after the P element:

If the page break type is "cloneall" - the three DIV elements will all be closed, by drawing the border and padding for each at the end of the page; the three DIV elements will be re-opened, drawing the borders and padding, at the top of the next page.

If the page break type is "clonebycss" - starting from the innermost element (div.level3) the DIV elements will have a border and padding at the end of the page if "box-decoration-break" is clone. In this case level2 and level 3 will be closed/cloned and level 1 will be sliced; the opposite will occur at the top of the next page.

Control of page breaks

Automatic page breaks (in flow of text)	Always "slice"
<tocpagebreak>	Always "cloneall"
<formfeed>	Always "slice"
If using columns	Always "cloneall"
Page break forced by change of @page selector	Always "cloneall"

<pagebreak>	Always "cloneall" if a change in page size or margins is specified. Otherwise page break type is determined by value of configurable variable: $this->defaultPagebreakType. Default is "cloneall". Default can be overridden by attribute "page-break-type" e.g. <pagebreak page-break-type="clonebycss" />
Page breaks forced by: page-break-before or page-break-after	Page break type determined by value of configurable variable: $this->defaultPagebreakType. Default is "cloneall".

Notes on page breaking

"box-decoration-break: slice | clone" was proposed for CSS3 in http://www.w3.org/TR/2012/CR-css3-background-20120417/#the-box-decoration-break but it appears that it may be withdrawn. Default is "slice"; it is not inherited.

"page-break-before" is not supported on <table>.

"page-break-before|after" is ignored if set on block elements inside a table.

mPDF functions e.g. AddPage() are not affected by the changes in mPDF 6.

Background images and gradients are not sliced.

$this->restoreBlockPagebreaks in config.php is now redundant.

Line breaking

The algorithm for determining automatic line breaks has been completely rewritten, ignoring inline tags (except for some cases of CJK line-breaking, and autohyphenation).

Line breaks will be allowed at:

- Spaces U+0020
- Word break U+200B
- Hyphen-minus U+002D when CSS hyphens set to "manual" or "auto", except when in a URL, or when following character is a > or numeral
- Hard hyphen U+2010 when CSS hyphens set to "manual" or "auto"
- Soft hyphen U+00AD "" when CSS hyphens set to "manual" or "auto"
- Automatic hyphenation when CSS hyphens set to "auto"
- Between CJK characters, except CJK numerals, before "CJK-following" or after "CJK-leading" characters

See also "Dictionary Line breaking" above.

Line-height and Text Baseline

Using font metrics

mPDF 6 can (optionally) use font metrics derived from each font file to:

- Determine the height of a line when line-height is set to 'normal'
- Determine the glyph baseline (previously a fixed value)

Options are set by configurable variables in the `config.php` file:

Default settings in mPDF versions 6 - recommended especially for complex scripts with marks used above or below characters:

- `$this->useFixedNormalLineHeight = false;`
- `$this->useFixedTextBaseline = false;`
- `$this->adjustFontDescLineheight = 1.14;`

Settings to be backwards compatible with mPDF versions < 6:

- `$this->useFixedNormalLineHeight = true;`
- `$this->useFixedTextBaseline = true;`
- `$this->normalLineheight = 1.33;`

Examples

Using the font metrics will give approximately the same result as the fixed value for many standard Latin script fonts e.g. DejaVu Sans Condensed:

line-height: normal; based on font metrics

line-height: normal; using fixed value

However, for some fonts the normal line-height using font metrics will be significantly taller, to account for the design of the font glyphs e.g. Khmer font:

line-height: normal; based on font metrics; ង្គ្រ ង្រ្គ ន្ត្រី ន្ត្រី ង្គ្រោះ យុវជន

line-height: normal; using fixed value; ង្គ្រ ង្រ្គ ន្ត្រី ន្ត្រី ង្គ្រោះ យុវជន

For more information on how complex normal lineheights are, see Eric Meyers' website: http://meyerweb.com/eric/thoughts/2008/05/06/line-height-abnormal/ and http://typophile.com/node/13081

CSS control of line-height

There are also new controls for line-height using draft CSS3 properties. These can be set on all block level elements (P, DIV etc) and tables (TABLE/TD/TH).

```
line-stacking-strategy = inline-line-height | block-line-height | max-height | grid-height
```

- `inline-line-height` - [default] lineheight is initially calculated from the block-level font[-size]; the height is expanded by any inline content, including the calculated lineheight of that inline content;
- `block-line-height` - lineheight is fixed as the lineheight of the block-level font[-size];
- `max-height` - lineheight is initially calculated from the block-level font; the height is expanded by any inline content, EXCLUDING the calculated lineheight of that inline content;
- `grid-height` - lineheight is initially calculated from the block-level font; the height is expanded - AS MULTIPLES OF INITIAL LINEHEIGHT - by any inline content, EXCLUDING the calculated lineheight of that inline content;

Note: XSL has a similar property with the same name, which uses different but equivalent values: `line-height` instead of `inline-line-height`, `font-height` instead of `block-line-height`. It also uses `max-height`. The value `grid-height` is new to the CSS3 property.

Examples

line-height: normal; DejaVu Sans Condensed

line-height: normal; 16pt font-size Â with line-stacking-strategy: inline-line-height

line-height: normal; 16pt font-size Â with line-stacking-strategy: block-line-height

line-height: normal; 16pt font-size Â with line-stacking-strategy: max-height

line-height: normal; 16pt font-size Â with line-stacking-strategy: grid-height

```
line-stacking-shift = consider-shifts | disregard-shifts
```

This property determines whether to include or disregard the adjusted top- and bottom-edge of any characters that have a baseline-shift (e.g. superscript) when calculating lineheight.

Note: XSL has a similar property with a different name: `line-height-shift-adjustment` which uses the same values.

Examples

In the table below, the line-height is set to 1em throughout the table; line-stacking-shift is set as 'disregard-shifts' in the first row, and has default setting (consider-shifts) in the second row.

Normal text DejaVu Sans[53] Condensed Normal text DejaVu Sans[53] Condensed Normal text DejaVu Sans[53] Condensed Normal text DejaVu Sans[53] Condensed Normal text DejaVu Sans[53] Condensed Normal text DejaVu Sans[53] Condensed	Normal text DejaVu Sans Condensed Normal text DejaVu Sans Condensed Normal text DejaVu Sans Condensed Normal text DejaVu Sans Condensed Normal text DejaVu Sans Condensed Normal text DejaVu Sans Condensed
Normal text DejaVu Sans[53] Condensed Normal text DejaVu Sans[53] Condensed Normal text DejaVu Sans[53] Condensed Normal text DejaVu Sans[53] Condensed Normal text DejaVu Sans[53] Condensed Normal text DejaVu Sans[53] Condensed	Normal text DejaVu Sans Condensed Normal text DejaVu Sans Condensed Normal text DejaVu Sans Condensed Normal text DejaVu Sans Condensed Normal text DejaVu Sans Condensed Normal text DejaVu Sans Condensed

For more details see the CSS3 draft specification.

Note for Advanced users

There are actually three possible metrics that can be used in a TrueType font file. The differences are summed up quite well in this article at http://typophile.com/node/13081. mPDF will by default use the usWinAscent and usWinDescent values to determine a 'normal' line-height, with two variations:

- if either the usWinAscent or usWinDescent are greater than the font bounding box (yMin yMax), then the values are reduced to equal the yMin/yMax values. NB this works as a fix with Myanmar Text (Windows 8 version) to give a line-height normal that is equivalent to that produced in browsers.
- if the USE_TYPO_METRICS bit is set on fsSelection (OS/2 table), this is telling the font to use the sTypo values and not the usWinAscent values. NB this works as a fix with Cambria Math font to

give a normal line-height; at present, this is the only font I have found with this bit set; although note that MS WordPad and Windows FireFox browser use the big line-height from usWinAscent, whilst MS Word 2007 observes the fSelection value.

You can change the font metrics used by mPDF, by editing the defined constant (_FONT_DESCRIPTOR) at the top of the mpdf.php file:

- 'winTypo' uses sTypoAscender etc from the OS/2 table and is the one officially recommended - BUT
- 'win' use usWinAscent etc from OS/2 and in practice seems to be used most commonly in Windows environment; this is the default in mPDF;
- 'mac' uses Ascender etc from hhea table, and may be used to give results consistent with a Mac/OSX environment.

Finally, you can override values for Ascent, Descent and Leading for any specific font, by setting values in config_font.php e.g.

```
	"cambriamath" => array(
		'R' => "cambria.ttc",
		'useOTL' => 0xFF,
		'TTCfontID' => array(
			'R' => 2,
			),
		'Ascent' => 950,
		'Descent' => -222,
		'Leading' => 0,
		),
```

Note - The same values are used for all styles of the font-family. Descent values should be negative. All values should be given using a 1000 units per em scale, regardless of the UnitsPerEm used in the font design.

Notes

Remember that line-height for a TABLE has a default value (1.2) set in the config.php default CSS. This is left in for backwards compatability. You can change this value to 'normal' for results consistent with most browsers.

Line-height in a <textarea> is fixed and defined in `classes/mpdfform.php` (= 1.2)

Details of the font metrics can be seen by inspecting the temporary font files e.g. `/ttfontdata/[fontname].mtx.php`.

Indexes

Index style and layout

Indexes have been completely rewritten for mPDF 6, and are not backwards compatible:

- Reference() is now removed - use IndexEntry() instead.
- CreateReference() and CreateIndex() are both removed - replaced by: InsertIndex() [or recommend <indexinsert>] cf. below.
- <indexinsert> and InsertIndex() no longer set styles - appearance must be controlled using CSS, even if using function InsertIndex().
- <indexinsert> and InsertIndex() no longer control columns - these must be specified separately.

When an Index is inserted in the PDF document, the Index is now generated (internally) as HTML code in the following format:

```
<div class="mpdf_index_main">
<div class="mpdf_index_letter">A</div>
<div class="mpdf_index_entry">Aardvark<a class="mpdf_index_link"
href="#page37">37</a>
</div>
...
</div>
```

CSS stylesheets can thus be used to control the layout of the Index e.g.:

```
/* For Index */
div.mpdf_index_main {
    line-height: normal;
    font-family: sans-serif;
}
div.mpdf_index_letter {
    line-height: normal;
    font-family: sans-serif;
    font-size: 1.8em;
    font-weight: bold;
    text-transform: uppercase;
    page-break-after: avoid;
    margin-top: 0.3em;
    margin-collapse: collapse;
}
div.mpdf_index_entry {
    line-height: normal;
    font-family: sans-serif;
    text-indent: -1.5em;
}
a.mpdf_index_link {
    color: #000000;
    text-decoration: none;
}
```

A default stylesheet for Indexes is included in mpdf.css (see note later for more information).

Index Collation

In order to generate an Index with non-ASCII characters, entries need to be sorted accordingly (collation), and non-ASCII characters should map to the appropriate Dividing letter e.g.:

A

Alonso, Fernando
Ãlvarez, Isaac
Arroyo Molino, David

B

BenÃtez, Carlos

Entries in an Index can now be sorted using any of the Locale values available on your system. Set it using the "collation" property/parameter e.g.:

```
<indexinsert usedivletters="on" links="off" collation="es_ES.utf8" collation-
group="Spanish_Spain" />
- or -
$mpdf->InsertIndex(true, false, "es_ES.utf8", "Spanish_Spain");
```

NB You should always choose a UTF-8 collation, even when you are using Core fonts or e.g. charset-in=win-1252, because mPDF handles all text internally as UTF-8 encoded.

You can see which Locales are available on your (Unix) system: `<?php system('locale -a') ?>`

Note: Index collation will probably not work on Windows servers because of the problems setting Locales under Windows.

If you have set your index to use Dividing letters, you can also determine how letters are grouped under a dividing letter. In the example index above, we want Ã to be grouped under the letter a/A. Set the "collation-group" using:

```
<indexinsert usedivletters="on" links="off" collation="es_ES.utf8" collation-
group="Spanish_Spain" /> - or -
$mpdf->InsertIndex(true, false, "es_ES.utf8", "Spanish_Spain");
```

Values should be selected from the available file names in folder /collations/.

Note: This will not affect the overall order of entries, which is determined by the value of "collation".

Note: The groupings do not always match the order set by locale. This is because the data for collations has come from different sources. The files in /collations/ can be edited.

The array consists of [index]: unicode decimal value of character => unicode decimal value of character to group under: e.g. Ã [A tilde] (U+00C3) (decimal 195) => a (U+0061) (decimal 97). The target character should always be the lowercase form.

Non-ASCII chcracters in Index entries

Note: htmlspecials_encode should be used to encode the text of content in <indexentry> - although not when using $mpdf->IndexEntry().

Columns

Columns are no longer specified as part of the <indexinsert>, so a typical 2-column index might be produced by:

```
<pagebreak type="next-odd" />
<h2>Index</h2>
<columns column-count="2" column-gap="5" />
<indexinsert usedivletters="on" links="on" collation="en_US.utf8"
collationgroup="English_United_States" />
<columns column-count="1" />
```

Index Sub-entries

Index entries can contain sub-entries, separated by colons e.g.

```
<indexentry content="Mammals:elephants" />
```

A shorthand way of displaying subentries is set by default, which suppresses the main entry if > 1 subEntry. It can be disabled/enabled using the configurable variable $this->indexUseSubentries in config.php.

This is the default appearance, with $this->indexUseSubentries = false; -

Mammals 73
- elephants 142
- humans 173

Marsupials
- kangaroos 75
- wombats 86

Index entries can also include simple mark-up tags and/or more than one colon e.g:

```
<indexentry content="Mammals:&lt;b&gt;elephants&lt;/b&gt;: breeding" />
```

which appears as:

Mammals
- **elephants**: breeding 15

This is the appearance with $this->indexUseSubentries = false; -

Mammals 73
Mammals, elephants 142
Mammals, **elephants**: breeding 15
Mammals, humans 173
Marsupials, kangaroos 75
Marsupials, wombats 86

Customised appearance

Several variables set at beginning of function InsertIndex() in mpdf.php which could be changed to alter appearance of Index. e.g. spacer, and joiner characters.

Lists

Lists are now handled as for other block level tags, so you can apply any CSS properties usable on blocks (e.g. border, background, padding) to UL/OL and LI tags.

CSS property "list-style" is now handled properly as a shorthand, and there is full support for "list-style-image", "list-style-type", and "list-style-position".

There are two modes for lists in mPDF 6: "mpdf" mode and "browser" mode. Mode is set using the configurable variable $this->list_auto_mode in config.php

1) Browser mode gives the same display as most browsers. In this mode, the default list indentation is set by padding "0 auto" in the default CSS in config.php. "auto" equates to the value of $this->list_indent_default in config.php - this is a "magic" value for padding, which is applied to either left or right depending on directionality of the list (rtl/ltr).

2) mPDF mode gives results compatible with previous versions of mPDF. In this mode, the indentation is calculated differently: the outside edge of the list item is considered to be the outside edge of the bullet or number. For numbered lists, mPDF calculates the width of the largest number and this width is used to set the outside edge. The default list indentation of "auto" in mPDF mode is set by $this->list_indent_default_mpdf. This value is added to the automatic calculated indentation. For backwards compatibility, $this->list_indent_first_level = 0; can be used to prevent any indentation of the first list level.

The automatic indentation only works for bullets or numbered lists, and is ignored if "list-style-position: inside" is set, or images are used for markers.

Browser mode is set as the default - for backwards compatibility, change this to "mpdf".

List top & bottom margins

The default in browsers is to add a top and bottom margin to the outermost list only. This can be defined using CSS as:

```
ul, ol { margin-top: 0.83em; margin-bottom: 0.83em; }
ul ul, ul ol, ol ul, ol ol { margin-top: 0; margin-bottom: 0; }
```

This style is included in file mpdf.css (see later).

Previous versions of mPDF always added a top and bottom margin to the outermost list, (but no variation from this was possible). mPDF 6 is therefore backwards compatible re. the margins.

[NB The CSS styles are included in mpdf.css, because the defaultCSS values set in config.php only works on basic elements, and cannot use selectors such as "ol ol".]

Other new configurable variables

Configurable variables are used to define size and offset for list bullets (i.e. disc, circle or square). The values can be any valid CSS size.

To specify a fixed bullet size and offset to give a similar appearance to most browsers, the default is set as:

- $this->list_marker_offset = '5.5pt';
- $this->list_symbol_size = '3.6pt';

To specify size and offset proportional to the list item's font size (compatible with previous versions of mPDF), use:

- $this->list_marker_offset = '0.45em';
- $this->list_symbol_size = '0.31em';

Notes on Lists

The attribute type="" is case sensitive (whereas it is case insensitive in CSS). This allows the use of shorthand versions e.g. type="A" for uppercase alpha-numeric.

"list-style-type" is only inherited to child LI (not to child UL/OL); list-style-image and -position are fully inherited.

Lists in tables remain basic, as block-level elements are not supported inside tables.

Properties like text-align:justify will now be inherited from surrounding elements, which will change the appearance of lists designed with earlier versions of mPDF.

The attribute start="3" (integer) works for "OL"; it is an official (though depracated) HTML attribute.

List bullets (type = disc, circle or square) are now drawn rather than using font glyphs, for better consistency.

List examples

This demonstrates the appearance when list_auto_mode is set to 'mpdf', compatible with previous mPDF versions. Indentation is set to zero (list_indent_default_mpdf). Note the top and bottom margin on the first list level only.

I. First item
II. Second item
    I. Next list level
    II. Second item

This demonstrates the same default settings, but list numbering is set to start at 32. Note how the indentation is adjusted to fit the maximum width of the numbering.

XXXII. First item
XXXIII. Second item
    XXXII. Next list level
    XXXIII. Second item

This demonstrates the appearance when list_auto_mode is set to 'browser', compatible with browsers. Indentation is set to 40px (list_indent_default)

I. First item
II. Second item
    I. Next list level
    II. Second item

This demonstrates the same as the previous example, but list numbering is set to start at 32. Note that the default indentation remains fixed at 40px (list_indent_default)

XXXII. First item
XXXIII. Second item
    XXXII. Next list level
    XXXIII. Second item

This demonstrates control of the list-style-type, list-style-position and list-style-image.

1. First item in list
2. Second item

- List style set as 'disc'

List style set as 'none'

➪ Using an image.

6. List-style-position: inside.

☺ User defined list bullet

Image transform

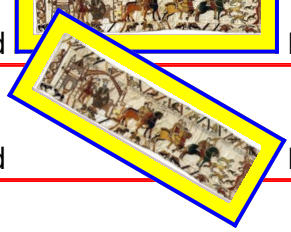
The CSS property "transform" is supported on images (only). All transform functions are supported except matrix() i.e. translate(), translateX(), translateY(), skew(), skewX(), skewY(), scale(), scaleX(), scaleY(), and rotate()

Transforms cannot be used when using columns or Keep-with-table (use_kwt).

The CSS property background-color is now supported on images.

In the following examples, note the difference between transform (which is applied after layout) and image-orientation (which is applied before layout):

Hallo world Hallo world

Hallo world  Hallo world

Hallo world Hallo world

Hallo world Hallo world (image-orientation)

CSS overline

The CSS property "text-decoration: overline" is supported. Note that since mPDF 5.7.3 text-decoration use the parent inline block baseline/fontsize/color for child inline elements, and allows nested use of these values and superscript/subscript.

1st place and compare with 1st

He won 1st place

Headers and Footers

Headers and Footers are all now written internally as HTMLheaders/footers. The use of non-HTML headers and footers is depracated, but remains supported. Non-HTML headers and footers are converted in mPDF to HTML equivalents.

Layout: This may mean that there will be a change in the resulting PDF. The main change is that an HTML table is created with three cells for left, right and middle; if you had a very long Left header item, it will not overwrite the center item, but it may wrap center onto 2 lines.

Naming: Default non-HTML headers will not clash with HTML headers, but named non-HTML headers WILL clash with (and overwrite) HTML headers of the same (equivalent) name e.g. html_MyFooter == MyFooter (non-HTML).

Aliases: {nb} or {nbpg} now only work in Headers or Footers, and not in the main text. {PAGENO} and {DATE ...} continue to only work in Headers or Footers.

ToC: Can now set the pagenumbering/style/reset/suppress for the ToC separately (see section on ToC).

The following are all depracated (but still supported) in favour of HTMLheader/footers:

- SetHeader()
- SetFooter()
- <pageheader>
- <pagefooter>
- DefHeaderByName()
- DefFooterByName()
- <setpageheader>
- <setpagefooter>
- SetHeaderByName()
- SetFooterByName()

If document direction is RTL (body dir=rtl, html dir=rtl), then you need to set directionality before setting non-HTML headers e.g.

```
$mpdf = new mPDF();
$mpdf->SetDirectionality('rtl'); // i.e. add this in
$mpdf->SetHeader($h);
$mpdf->SetFooter($f);
$mpdf->WriteHTML('<body dir="rtl">...');
```

Table of Contents (ToC)

Page numbering can now be applied and controlled for the pages containing a ToC.

There are three new parameters to control pagenumbering in the ToC: toc-resetpagenum, toc-pagenumstyle, and toc-suppress. These are set as attributes in <tocpagebreak> or as the last 3 parameters in TOCpagebreak(); they set the pagenumbering and pagenumbering style for the ToC, and whether to suppress pagenumbers in ToC.

The default setting for all is to continue pagenumbering and pagenumstyle (and suppression) from pages preceding the ToC.

Note: Page numbering will **always** reset following a ToC. By default it will set it to 1, unless a value for resetpagenum is specified in TOCpagebreak or <tocpagebreak>.

Backwards compatibility: page numbers are no longer suppressed by default in ToC.

Although "suppress" and "toc-suppress" are supported, the recommended way to control whether page numbering appears is by using different headers and footers for each section.

Note: If you have 2 ToCs immediately following each other, and wish to use pagenumstyle or suppress to control the following text, then you need to set those values on both of the <tocpagebreak> elements.

The default CSS styles for ToCs and Indexes are now set in mpdf.css (see later).

See notes later on page numbering.

Other changes from mPDF 5

Setting up mPDF 6

mPDF 6 has changed significantly from earlier version and it is recommended that a fresh install is used. You may wish to copy your previous config_* files and use them to update the new config files.

config_fonts.php - values of "indic" and "unAglyphs" from previous versions are now redundant.

config_lang2fonts.php - this is similar to the previous config_cp.php file; note however that $unifont (NOT $unifonts) must be only one font (not a comma-separated list as before).

Included fonts - the Indic fonts e.g. ind_bn_001.ttf are no longer required (nor do they work properly with mPDF 6).

useLang - this configurable variable, which used to be true by default, is now redundant. You may need to set: $mpdf->autoLangToFont = true; for the same results.

SetAutoFont() - is now redundant. You may need to set: $mpdf->autoScriptToLang = true; for the same results.

Indexes - have been largely redefined. See the section above.

Lists - have been rewritten. See the section above.

Headers and Footers - have been rewritten. See the section above.

A number of old depracated aliases will no longer be supported. Warning errors have been added to prompt you to change to the updated form:

- $mpdf->useOddEven - should now use - $mpdf->mirrorMargins
- $mpdf->useSubstitutionsMB - should now use - $mpdf->useSubstitutions
- $mpdf->AliasNbPg - should now use - $mpdf->aliasNbPg
- $mpdf->AliasNbPgGp - should now use - $mpdf->aliasNbPgGp
- $mpdf->BiDirectional - should now use - $mpdf->biDirectional
- $mpdf->Anchor2Bookmark - should now use - $mpdf->anchor2Bookmark
- $mpdf->KeepColumns - should now use - $mpdf->keepColumns
- $mpdf->UnvalidatedText - should now use - $mpdf->watermarkText
- $mpdf->TopicIsUnvalidated - should now use - $mpdf->showWatermarkText
- $mpdf->Reference - should now use - $mpdf->IndexEntry

The following functions have been removed:

- setUnvalidatedText - should now use - SetWatermarkText()
- AddPages - should now use - AddPage() or HTML code methods
- startPageNums
- CreateReference and CreateIndex - cf. Index section above

Default style sheet

A new mpdf.css file includes defaults for LISTS top/bottom margins, and also examples for Indexes and ToCs. This now acts like a normal CSS file, including cascading selectors i.e. not just main tags. This is always read (if present), so acts as a secondary default CSS, but one which allows selectors. Styles added to this act like a user stylesheet when considering precedence e.g. cellSpacing and border-spacing.

Direct writing methods and OTL

WriteText() WriteCell() Watermark() AutoSizeText() and ShadedBox() DO support complex scripts and right-to-left text (RTL).

Write() does NOT support complex scripts or RTL (NB this is a change - Write() used to support RTL).

CircularText() does NOT support complex scripts or RTL.

MultiCell() DOES support complex scripts and RTL, but complex-script line-breaking MAY NOT be accurate.

MultiCell() does not support kerning and justification. NB This includes <textarea> in forms which uses MultiCell() internally.

<select> form objects also do NOT support kerning.

Page numbering

Page numbering i.e. by including {PAGENO} or {nbpg} in a header/footer, can use any of the number types as used for list-style e.g.

```
<pagebreak pagenumstyle="arabic-indic">
```

Short codes are recognised for the 5 most common:

- "1" - decimal
- "A" = upper-latin or upper-alpha
- "a" = lower-latin or lower-alpha
- "I" = upper-roman
- "i" = lower-roman

or any of the following: arabic-indic, hebrew, bengali, devanagari, gujarati, gurmukhi, kannada, malayalam, oriya, persian, tamil, telugu, thai, urdu, cambodian, khmer, lao, cjk-decimal

Note: A suitable font must be used in the header/footer in order to display the numbers in the selected script.

You can now set the pagenumberstyle from the beginning of the document by changing the configurable variable:

```
$this->defaultPageNumStyle = "arabic-indic"; // in config.php
$mpdf->defaultPageNumStyle = "arabic-indic"; // at runtime
```

Other Minor changes in mPDF 6

'hebrew', 'khmer', 'cambodian', 'lao', and 'cjk-decimal' are recognised as values for "list-style-type" in numbered lists.

CSS "text-outline" is now supported on TD/TH tags

Text wrapping in tables has been improved when using CJK scripts (chinese-japanese-korean).

Text underline and strikethrough can be used together: ~~Hallo world~~. Either <u><s>...</s></u> or ... can be used

Added support for style="opacity:0.6;" in SVG - equivalent to: style="fill-opacity:0.6; stroke-opacity: 0.6;"

Added support for opacity="0.6" (as attribute) in SVG - previously only supported fill-opacity="0.6" stroke-opacity="0.6"

CSS position:absolute or fixed - rotate extended now to include rotate: 180; (previously just 90 or -90)

The default value of $this->keep_table_proportions = true; in config.php has been changed (see effect on Example 6 - nested table in top right cell).

Limited support has been added for SVG fonts embedded in SVG images (but not using @font-face rules) - see the separate Images demo file.

When using columns, the top margin is now collapsed at top of every column (not just first column of page).

The way mPDF handles optional end tags has been updated to be consistent with the HTML5

specification - previously not well defined for HTML4.

Changes to the way lists are handled means that text-align:justify may be inherited by lists from surrounding block elements (which did not happen previously). See LISTS above for more information.

Backwards Compatibility

For maximum backwards comaptibility with older versions of mPDF, change the following configurable variables in the `config.php` file:

	mPDF 6.0 Default (Browser compatible)	Backwards Compatible
Normal Line-height	$this->useFixedNormalLineHeight = false; $this->useFixedTextBaseline = false; $this->adjustFontDescLineheight = 1.14;	$this->useFixedNormalLineHeight = true; $this->useFixedTextBaseline = true; $this->normalLineheight = 1.33;
Lists	$this->list_auto_mode = 'browser'; $this->list_marker_offset = '5.5pt'; $this->list_symbol_size = '3.6pt';	$this->list_auto_mode = 'mpdf'; $this->list_marker_offset = '0.45em'; $this->list_symbol_size = '0.31em';

More Information

For more information, see:

- About OTL: http://www.microsoft.com/typography/otspec/TTOCHAP1.htm
- OTL tag Registry: http://www.microsoft.com/typography/otspec/ttoreg.htm
- OTL Features list: http://www.microsoft.com/typography/otspec/featurelist.htm
- CSS3 Font Features: http://dev.w3.org/csswg/css-fonts/#font-rend-desc

Font Information

The following fonts are included with mPDF 6:

Font(s)	Download URL	Copyright / License	Coverage
DejaVuSans DejaVuSansCondensed DejaVuSerif DejaVuSerifCondensed DejaVuSansMono	http://dejavu-fonts.org	© Bitstream http://dejavu-fonts.org/wiki/License	[Numerous]
FreeSans FreeSerif FreeMono	http://www.gnu.org/software/freefont/	GNU GPL v3	[Numerous incl. Indic]
Quivira	http://www.quivira-font.com/	*free for any use*	Coptic Buhid Tagalog Tagbanwa Lisu
Abyssinica SIL	http://www.sil.org/resources/software_fonts/abyssinica-sil	SIL Open Font License	Ethiopic
XBRiyaz	http://www.redlers.com/downloadfont.html (XW Zar fonts) http://wiki.irmug.org/index.php/XWZar	SIL Open Font License	Arabic
Taamey David CLM	http://opensiddur.org/tools/fonts/	GNU GPL 2	Hebrew
Estrangelo Edessa	http://www.bethmardutho.org/index.php/resources/fonts.html (SyrCOMEdessa.otf)	Adapted licence (free to use/share)	Syriac
Aegean	http://users.teilar.gr/~g1951d/	*free for any use*	Carian Lycian Lydian Phoenecian Ugaritic Linear B Old Italic
Jomolhari	https://sites.google.com/site/chrisfynn2/home/fonts/jomolhari	SIL Open Font License	Tibetan
Lohitkannada	https://fedorahosted.org/lohit/	SIL Open Font License	Kannada
Kaputaunicode	http://www.kaputa.com/slword/kaputaunicode.htm http://www.locallanguages.lk/sinhala_unicode_converters	Free Sri Lanka Web Community Center	Sinhala
Pothana2000	https://fedoraproject.org/wiki/Pothana2000_fonts	GNU GPL v2+	Telugu
Lateef	http://www.sil.org/resources/software_fonts/lateef	SIL Open Font License	Sindhi
Khmeros	http://www.khmeros.info/en/fonts (http://www.cambodia.org/fonts/)	LGPL Licence	Khmer
Dhyana	Google Fonts http://www.google.com/fonts/earlyaccess	SIL Open Font License	Lao
Tharlon	Google Fonts http://code.google.com/p/tharlon-font/	SIL Open Font License	Myanmar Tai Le
Padauk Book	http://www.sil.org/resources/software_fonts/padauk	SIL Open Font License	Myanmar
Ayar fonts	http://eng.ayarunicodegroup.org/	SIL Open Font License	Myanmar
ZawgyiOne	http://code.google.com/p/zawgyi/wiki/MyanmarFontDownload	Freely available. No licence information available	Myanmar
Garuda	http://www.hawaii.edu/thai/thaifonts/	Freely available. No licence information available	Thai
Sundanese Unicode	http://sabilulungan.org/aksara/	GNU GPL	Sundanese
Tai Heritage Pro	http://www.sil.org/resources/software_fonts/tai-heritage-pro	SIL Open Font License	Tai Viet
Sun-ExtA Sun-ExtB	http://www.alanwood.net/downloads/index.html	Freeware (Beijing ZhongYi Electronics Co)	Chinese Japanese Runic
Unbatang	http://kldp.net/projects/unfonts/download	GNU GPL	Korean
Aboriginal Sans	http://www.languagegeek.com/font/fontdownload.html	GNU GPL 3	Cree Canadian Aboriginal Inuktuit

Font(s)	Download URL	Copyright / License	Coverage
MPH 2B Damase	http://www.alanwood.net/downloads/index.html	(Public domain)	Glagolitic Shavian Osmanya Kharoshthi Deseret
Aegyptus	http://users.teilar.gr/~g1951d/	*free for any use*	Egyptian Hieroglyphs
Akkadian	http://users.teilar.gr/~g1951d/	*free for any use*	Cuneiforn
Eeyek Unicode	http://tabish.freeshell.org/eeyek/download.html	Freeware	Meetei Mayek
Lannaalif	http://www.geocities.jp/simsheart_alif/taithamunicode.html	(Unclear)	Tai Tham
Daibanna SIL Book	http://www.sil.org/resources/software_fonts/dai-banna-sil	SIL Open Font License	New Tai Lue
KFGQPC Uthman Taha Naskh	http://fonts.qurancomplex.gov.sa/?page_id=42	https://www.ohloh.net/licenses/KFGQPC	Arabic (Koran/Quran)