Das microtype Paket

Eine Schnittstelle für die mikro-typographischen Erweiterungen von pdfTeX

Autor: R. Schlicht (w.m.l@gmx.net)
Version: v2.4 (2010/01/10)

Die **Übersetzung ins Deutsche** wurde von Jenny Schröder (JennySchroeder002@web.de) und Sebastian Schwerdtfeger (sebastian.schwerdtfeger@uni-jena.de), Studierende der Friedrich-Schiller-Universität Jena, verfasst.

Zusammenfassung

Das microtype Paket stellt eine LaTeX Schnittstelle für die mikrotypographischen Erweiterungen von pdfTeX bereit: besonders markant Zeichenvorsprünge und Fontexpansion, weiterhin die Justierung von Zeichenabständen innerhalb Wörter und zusätzliches Kerning sowie durch Bindestrich trennbaren Sperrsatz (Tracking) und die Möglichkeit, alle oder ausgewählte Ligaturen abzuschalten. Es erlaubt, diese Features auf anpassbare Fonts anzuwenden und alle mikrotypographischen Aspekte dieser Fonts in einer einfachen und flexiblen Art und Weise zu konfigurieren. Einstellungen für verschiedene Schriftarten werden bereitgestellt.[1]

Zu beachten ist, dass Fontexpansion und Zeichenvorsprünge nur mit pdfTeX ($\geq$ Version 0.14f) arbeiten. Automatische Fontexpansion erfordert mindestens Version 1.20 oder neuer. Ligaturen zu deaktivieren erfordert pdfTeX 1.30, Sperrsatz und die Justierung von Zeichenabständen innerhalb Wörter, sowie Kerning erfordern Version 1.40. Das Paket aktiviert standardmäßig Protrusion und Expansion, falls sicher angenommen werden kann, dass diese funktionieren. Diese beiden Features sind ebenso verfügbar mit luaTeX. Das microtype Paket arbeitet nicht mit XƎTeX.

Das alternative Paket letterspace, welches ebenso mit einfachem TeX arbeitet, stellt nur die Befehle für Sperrsatz (*letterspacing*) bereit und lässt Support für alle anderen Erweiterungen aus (siehe Abschnitt 7).

---

[1] Zur Zeit stellt dieses Paket Protrusionseinstellungen für Computer Modern Roman, Palatino, Times, URW Garamond, Adobe Garamond und Minion, Bitstream Charter und Letter Gothic, die AMS Symbole und Euler Fonts, verschiedene Eurosymbol Schriftarten, sowie einige allgemeine Einstellungen für unbekannte Schriftarten (siehe Tabelle 3 auf Seite 23) bereit. Beisteuerungen sind gern gesehen.

Inhaltsverzeichnis

Tabellenverzeichnis

1 Mikrotypographie mit pdfTEX

pdfTEX, die TEX Erweiterung, geschrieben von Hàn Thế Thành, führt eine Reihe mikrotypographischer Features ein, die es nicht nur zum Werkzeug der Wahl für die Erstellung elektronischer Dokumente, sondern auch für Werke überragender, altehrwürdiger Typographie macht: am bedeutendsten Zeichenvorsprünge (Character Protrusion, auch bekannt als margin kerning) und Fontexpansion. Hàn Thế Thànhs führte aus:

'Margin kerning bezeichnet die Justierung der Zeichen an den Rändern (margin) eines gesetzten Textes. Eine vereinfachte Anwendung von Margin kerning ist eine hängende Zeichensetzung. Margin kerning ist notwendig für optische Ausrichtung der Ränder eines gesetzten Textes, weil mechanisches Angleichen der Ränder diese eher stümperhaft aussehen lässt. Einige Zeichen können eine Zeile für das menschliche Auge kürzer erscheinen lassen als andere. Solche Zeichen, um einen passenden Betrag zu den Rändern zu verschieben, würde das Aussehen des Textes massiv verbessern.

Mit Fontexpansion zu setzen ist die Methode, eine weitere oder engere Variante des Fonts zu erschaffen, welche die Wortzwischenräume besser ausgleicht. Eine Schriftart in einer losen Zeile kann durch eine weitere Variante ausgetauscht werden, sodass die Wortzwischenräume weniger stark gestreckt werden. Gleichermaßen kann ein Font in einer dicht bepackten Zeile durch eine schmalere Variante ersetzt werden, um die Abstände zu verringern. Es besteht mit Sicherheit die potentielle Gefahr der Fontdeformierung, wenn solche Manipulationen vorgenommen werden, weshalb sie mit extremer Vorsicht vorgenommen werden müssen. Das Potential, eine Zeilenbreite mittels Fontexpansion zu justieren, sollte bedacht werden, wenn ein Paragraph in Zeilen zerbrochen wird, um bessere Stopppunkte zu erhalten.' [Thành 2000, p. 323]

Diesen beiden Features fehlte seit einiger Zeit eine einfache LATEX Benutzerschnittstelle. Dann wurde das pdfcprot Paket veröffentlicht, welches LATEX Benutzern ermöglichte, Zeichenvorsprünge zu benutzen, ohne sich zu sehr mit den Interna beschäftigen zu müssen.

Fontexpansion allerdings war weiterhin schwierig zu utilisieren, da es erforderte, dass die Fontmetriken für alle Ebenen der Erweiterung verfügbar waren. Deswegen musste jeder, der von diesem Feature Gebrauch machen wollte, im Vorfeld multiple Ausführungen der Schriftarten anfertigen. Shellskripte, um den Benutzer wenigstens teilweise von dieser Last zu befreien, waren verfügbar – dennoch blieb es eine lästige Arbeit. Darüber hinaus mussten alle Fonts weiterhin physisch erstellt werden, wodurch Kompilierungszeit und Speicherplatz auf der Festplatte verschwendet wurden.

Im Sommer 2004 implementierte Hàn Thế Thành eine Besonderheit, welche sich als eine bedeutende Erleichterung für TEX- und LATEX-Benutzer entpuppte: Fontexpansion kann nun automatisch stattfinden. Das heißt, pdfTEX braucht nicht länger die Fontmetriken der expandierten Schriftart, sondern berechnet diese zur Laufzeit und komplett im Speicher.

Nach diesem großen Sprung in der Anwendbarkeit stoppte die Entwicklung nicht. Im Gegenteil, pdfTEX wurde mit immer mehr besonderen Merkmalen ausgestattet: Version 1.30 führte die Möglichkeit ein, *alle Ligaturen auszuschalten*, Version 1.40 brachte ein robustes *Letter-spacing*-Kommando, die Möglichkeit *zusätzliches Zeichen Kerning* zu spezifizieren und die *Justierung von Zeichenabständen innerhalb Wörter*.

Robustes und per Bindestrich trennbares *Letter-spacing (Tracking)* war immer extrem schwierig in TEX zu realisieren. Obwohl das soul Paket große Mühen unternahm, um es möglich zu machen, konnte es trotzdem weiterhin in bestimmten Situationen scheitern; allein das Tracking einer Schriftart für das ganze Dokument blieb unmöglich. Wendet man die neue Erweiterung von pdfTEX an, stellt dies nicht länger ein Problem dar. Das microtype Paket stellt die Möglichkeit bereit, das Tracking anpassbarer Fontsets zu ändern, z. B., alle kleinen Kapitälchen. Es führt außerdem die zwei neuen Kommandos \textls und \lsstyle für ad-hoc Letter-spacing ein, welche wie normale Textkommandos benutzt werden können. Beachten Sie, dass Letter-spacing nur im PDF-Modus funktioniert.

Zusätzliches kerning für Zeichen einer Schriftart einzustellen ist besonders nützliche für Sprachen, deren typographische Tradition es erfordert, bestimmte Zeichen durch ein Leerzeichen abzutrennen. Es ist zum Beispiel üblich, in französischer Typographie ein wenig Platz vor einem Fragezeichen, Ausrufezeichen und Semikolon zu lassen, und ein wenig mehr Platz vorm Doppelpunkt und den Guillemets. Bis jetzt konnte das nur erreicht werden, indem man diese Zeichen aktivierte (z. B. mit dem babel-Paket), was nicht immer eine praxistaugliche Lösung sein kann. Im Kontrast zum Standard Kerning, welches in die Schriftarten eingebaut ist (und welches natürlich wie üblich greift), basiert dieses Kerning auf einzelnen Zeichen nicht auf Zeichenpaaren.

Wortzwischenräume zu justieren basiert auf der Idee, dass – um eine uniforme Grauheit in einem Text zu erreichen – die Wortzwischenräume auch von den Zeichen in der direkten Umgebung abhängen sollten. Wenn zum Beispiel ein Wort mit einem 'r' endet, sollte der folgende Abstand ein kleines bisschen kleiner sein als z. B. nach einem 'm'. Man kann von diesem Konzept wie von einer Erweiterung von TEXs 'space factors' denken. Allerdings, während space factors alle drei Parameter dieser Wortzwischenräume (oder Haftung) um den selben Betrag beeinflusst – das Kerning, den maximalen Betrag, um den der Zwischenraum gestreckt oder geschrumpft werden kann – stellt pdfTEX die Möglichkeit bereit, diese Parameter unabhängig voneinander zu modifizieren. Darüber hinaus können die Werte für jeden Font anders eingestellt werden. Außerdem – und vielleicht am wichtigsten – können diese Parameter nicht nur erhöht sondern auch erniedrigt werden. Dieses Merkmal wird das Aussehen von Paragraphen vielleicht noch mehr aufwerten. Die Betonung liegt im letzten Satz auf dem Wort 'vielleicht': Diese Erweiterung ist immer noch sehr experimentell – speziell haben nur die letzten Zeichen im Moment Einfluss auf die Wortzwischenräume. Außerdem sind die mit microtype gelieferten Einstellungen nicht mehr als eine erste Annäherung, weshalb ich es begrüßen würde, wenn Sie Korrekturen und Verbesserungen vornehmen. Ich würde vorschlagen, die Begründung für die Einstellungen in Abschnitt 15.9 durchzulesen.

Abschließend mag die Möglichkeit, für einen Font *alle Ligaturen auszuschalten*, nützlich für Schreibmaschinenschriftarten sein.

Das microtype Paket stellt für all diese mikrotypographischen Erweiterungen eine Schnittstelle bereit. Alle mikrotypographischen Aspekte können individuell und einfach angepasst werden. Die nächsten Kapitel werden einen Abriss aller Optionen und Anpassungsmöglichkeiten präsentieren.

2 Loslegen

Es gibt keine Überraschungen beim Laden des Paketes:

```
\usepackage{microtype}
```

In den meisten Fällen ist das ausreichend, und falls Sie nicht an der Feinabstimmung des mikrotypographischen Aussehens ihres Dokumentes (was unwahrscheinlich erscheint, da Sie dieses Paket benutzen, was ihr Interesse an typographischen Problemen bestätigt) interessiert sind, können Sie den Rest dieses Dokumentes auslassen. Falls Sie dies andererseits nicht befriedigt – sei es aus theoretischen oder praktischen Gründen – wird Sie dieses Handbuch auf den Weg zu den erwünschten Ergebnissen und folgenden Meilensteinen bringen:

- Das entsprechende mikrotypograpische Feature einschalten, entweder mittels der respektiven package-Option oder mittels des \microtypesetup-Kommandos (Abschnitt 3).

- Wählen Sie die Schriftarten aus, auf welche dieses Feature angewendet werden soll, indem Sie 'Font Sets' deklarieren und aktivieren. Einige Sets sind vorerstellt und können direkt in den Paketoptionen aktiviert werden (Abschnitt 4).

- Feinabstimmen der mikrotypographischen Einstellungen an den Schriftarten oder Sets von Schriftarten (Abschnitt 5).

- Falls sie der Typ sind, der immer weiter marschiert, werden Sie sicher an der Möglichkeit zu Context-abhängigen Einstellungen interessiert sein (Abschnitt 6).

- Wir ermutigen Sie sogar, den Pfad typographischer Tugend zu verlassen und ein paar Schafe zu stehlen (section 7) oder auf andere Weise einzudringen (section 8).

- Sollten sich Ihnen ein Hindernis auftun, folgen Sie den Tipps und Warnungen (Abschnitt 9).

3 Optionen

Wie viele andere LaTeX Pakete auch akzeptiert das microtype Paket Optionen über die bekannte Schlüssel=Wert-Syntax. Im Folgenden finden sie eine Beschreibung aller Schlüssel und ihrer möglichen Werte ('true' kann ausgelassen werden; multiple Werte (falls erlaubt) müssen eingeklammert werden; der Standardwert wird rechts gezeigt, angeführt von einem Sternchen, falls Abhängigkeit von der pdfTeX Version und/oder dem Output Modus besteht).

3.1 Die mikro-typographischen Features anschalten

<div style="float:left">protrusion</div> true, false, compatibility, nocompatibility, ⟨*Name des Font Sets*⟩   *true

<div style="float:left">expansion</div> Dies sind die hauptsächlichen Optionen um die Ebene mikro-typographischer Verfeinerung, welche die Schriftarten in ihrem Dokument erhalten sollten, zu kontrollieren. Standardmäßig ist das Paket recht gierig; Character Protrusion ist aktiviert, Fontexpansion wird nur in Situationen abgeschaltet, in denen pdfTEX die Fonts nicht automatisch expandieren kann, also falls es entweder zu alt ist (Versionen vor 1.20) oder falls der Output Modus DVI ist (siehe Abschnitt 3.5). Anders gesagt, wird microtype versuchen, soviel Mikrotypographie anzuschalten wie es sicher als funktionierend unter den gegebenen Bedingungen erwarten kann (und normalerweise ist es nicht nötig, das Paket mit anderen Optionen für die PDF bzw. DVI Modi zu laden).

Protrusion und Expansion können getrennt (de)aktiviert werden, indem man activate den betreffenden Schlüssel auf true bzw. false setzt. Die activate Option kürzt das Setzen beider Optionen zur gleichen Zeit ab. Deswegen haben die folgenden Zeilen alle denselben Effekt (wenn man PDF-Dateien mit einer aktuellen Version von pdfTEX erstellt):

```
\usepackage[protrusion=true,expansion]{microtype}
```

```
\usepackage[activate={true,nocompatibility}]{microtype}
```

```
\usepackage{microtype}
```

Wenn man pdfTEX-Fontexpansion und Character Protrusion verwendet, können Zeilenumbrüche (und konsequenterweise auch Seitenumbrüche) unterschiedlich ausfallen. Wenn das nicht gewünscht ist – weil sie z. B. ein Buch neu setzen, dessen Seitenumbrüche sich nicht ändern dürfen – können Sie den Wert compatibility an die protrusion und/oder expansion Optionen hängen. Aus typographischer Sicht jedoch werden die Ergebnisse suboptimal sein, weshalb der Wert standardmäßig nocompatibility ist.

Abschließend können Sie auch den Namen eines Font Sets angeben, auf welches Character Protrusion und/oder Fontexpansion beschränkt werden sollten, siehe auch den Abschnitt 4 für eine detaillierte Beschreibung. Das Spezifizieren eines Font Sets für ein Features aktiviert dieses Features implizit mit.

<div style="float:left">tracking</div> true, false, ⟨*font set name*⟩   false

<div style="float:left">kerning
spacing</div> Es existiert keine Kompatibilitätsebene für die neuen Erweiterungen Tracking, zusätzliches Kerning und Wortzwischenräume (interword spacing). Deshalb können sie nur an- oder ausgeschaltet werden, oder sie können aktiviert werden, indem man einen Setnamen an die Option leitet. Standardmäßig ist keines der Features angeschaltet.

In Tabelle 1 erhalten Sie einen Überblick über die verfügbaren und standardmäßig aktivierten mikro-typographischen Features für die relevanten pdfTEX Versionen und Output-Modi.

Ob Ligaturen deaktiviert werden sollten, kann nicht mittels einer Option kontrolliert werden, sondern per \DisableLigatures-Kommando, welches in Abschnitt 8 erklärt wird.

Tabelle 1: Verfügbarkeit mikro-typographischer Features

TeX engine			Mikrotypographische Features					
Engine	Version	Output	Protrusion	Expansion	(= auto)	Kerning	Spacing	Tracking
pdfTeX	< 0.14f	DVI/PDF	∅	∅	∅	∅	∅	∅
	≥ 0.14f	DVI/PDF	★	⊠	∅	∅	∅	∅
	≥ 1.20	DVI	★	⊠	∅	∅	∅	∅
		PDF	★	★	★	∅	∅	∅
	≥ 1.40	DVI	★	⊠	∅	⊠	⊠	∅
		PDF	★	★	★	⊠	⊠	⊠ [a]
luaTeX	≥ 0.25	DVI	★	⊠	∅	∅	∅	∅
		PDF	★	★	★	∅	∅	∅

★ = aktiviert   ⊠ = nicht aktiviert   ∅ = nicht verfügbar   a ≥ 1.40.4 empfohlen

3.2 Zeichenvorsprünge

factor ⟨integer⟩   1000

Benutzen Sie diese Option, um global einzustellen, um welchen Betrag die Zeichen hervorragen sollen. Während ein Wert von 1000 bedeutet, dass der volle Vorsprung, wie in der Konfiguration (siehe Abschnitt 5.1) spezifiziert, benutzt wird, bedeutet ein Wert von 500 entsprechend die Halbierung der Protrusion. Dies kann nützlich sein, falls Sie grundsätzlich zufrieden mit den Einstellungen sind, aber zu bevorzugen, wenn die Zeichenhervorragungen weniger oder mehr zu sehen sind. (z.B. wenn sie so stolz darauf sind, dieses Feature nutzen zu können, dass sie es jedem zeigen wollen, oder – um eine Motivation zu nennen, die in Einverständnis mit typographischer Korrektheit steht – falls sie eine große Schriftart nutzen, die gemäßigtere Vorsprünge verlangt).

unit character, ⟨dimension⟩   character

Diese Option wird in Abschnitt 5.1 erklärt, ebenso das Kommando \SetProtrusion. Seien sie vorsichtig bei Benutzung dieser Optionen!

3.3 Fontexpansion

auto true, false   * true

Wie bereits im Kapitel 1 angemerkt, können die expandierten Versionen der Fonts automatisch erstellt werden. Diese Option ist standardmäßig an, wenn pdfTeX-Version 1.20 oder höher beträgt und der Output-Modus PDF ist; ansonsten ist die Option deaktiviert. Falls auto auf false gesetzt ist, müssen die einzelnen Fonts aller Expansionsstufen bereits existieren (mit nach dem Schema ⟨font name⟩±⟨expansion value⟩ benannten Dateien, z.B. cmr12+10, wie beschrieben in pdfTeX manual).

Automatische Fontexpansion funktioniert nicht mit Bitmap Fonts, weswegen Sie, falls Sie die Computer Modern Roman Fonts in T1-Kodierung[2], nutzen, entweder

---

2 Beiläufig bemerkt, Type 1 Format und T1-Encoding hängen in keiner Weise miteinander zusammen, abgesehen davon, dass beide mit einem 'T' anfangen und mit einer '1' enden.

cm-super Fonts installieren oder die Latin Modern Fonts (package lmodern) benutzen sollten.

stretch ⟨*integer*⟩ 20

shrink Sie können bestimmen, wie sehr ein Font maximal gestreckt bzw. geschrumpft werden kann. Die Zahl wird durch 1000 dividiert, sodass ein Stretch Limit von 10 bedeutet, dass ein Font um bis zu 1% expandiert werden kann. Das standardmäßige Stretch Limit ist 20; das Shrink Limit ist standardmäßig gleich dem Stretch Limit.

step ⟨*integer*⟩ *1

Fonts werden nicht um beliebige Größen expandiert, sondern nur um gewisse, bestimmte Schritte innerhalb der Expansionslimits. Mit aktuellen Version von pdfTeX (1.40 oder neuer) ist diese Option standardmäßig auf 1 gesetzt, damit pdfTeX die maximale Anzahl an Instanzen der jeweiligen Fonts ausprobieren kann und somit den bestmöglichen Output garantiert.[3] Ältere pdfTeX Versionen jedoch mussten jede einzelne Fontinstanz in der PDF Datei einfügen, wodurch die Dateigröße teils dramatisch erhöht wurde. Deswegen – sollten Sie eine pre-1.40 pdfTeX Version nutzen – ist **step** standardmäßig auf ein Fünftel des kleineren Wertes von **stretch** und **shrink** gesetzt.

selected true, false false

Wenn Fontexpansion angewendet wird ist es möglich, die Expansion einiger sensibler Zeichen, welche schnell deformiert erscheinen können (z. B. das 'O' im Kontrast zum 'I'), zu begrenzen. Das nennt sich *selected expansion* und seine Benutzung erlaubt, die Stretch und Shrink Limits zu erhöhen (auf z. B. 30 anstelle von 20); allerdings ist der Zuwachs limitiert, da gleichzeitig die durchschnittliche Stretch Varianz verringert wird Deswegen ist diese Option standardmäßig auf **false** gesetzt, sodass alle Zeichen um denselben Betrag expandiert werden. Siehe Abschnitt 5.2 für eine detailliertere Beschreibung.

3.4 Tracking/Letterspacing

letterspace ⟨*integer*⟩ 100

Diese Option verändert den Standardbetrag fürs Tracking (siehe Abschnitt 5.3) sowie Letterspacing (siehe Abschnitt 7). Der Betrag wird in Tausendsteln von 1 em angegeben; zulässige Werte sind im Bereich von −1000 bis +1000.

3.5 Verschiedene Optionen

DVIoutput true, false *false

pdfTeX ist nicht bloß in der Lage, PDF-Dateien zu erstellen, sondern kann auch DVI- Dateien erzeugen.[4] Letztere können mit der Option **DVIoutput** erzwungen werden, was \pdfoutput auf Null setzt.

---

3 Die Kehrseite an dieser Standardeinstellung ist, dass pdfTeX vielleicht bei großen Dokumenten nicht genügend Speicher zur Verfügung hat; in diesem Falle sollten Sie über die Fehlermeldungen in der 'Hints und caveats'-Abschnitt (9) nachlesen oder es mit einem größeren **step** probieren.

4 Neuere TeX Systeme benutzen pdfTeX als Standard Engine auch für DVI-Ausgaben.

Beachten Sie, dass dies Pakete verwirren wird, die auf den Wert von `\pdfoutput` angewiesen sind, falls diese früher geladen wurden; so erwarten sie, dass sie aufgerufen wurden um eine PDF-Ausgabe zu erzeugen, obwohl das nicht der Fall ist. Diese Pakete sind u. a.: graphics, color, hyperref, pstricks und natürlich ifpdf. Entweder laden sie diese Pakete nach microtype oder benutzen die Anweisung `\pdfoutput=0` früher – in letzterem Fall ist die `DVIoutput`-Option redundant.

Wenn man DVI-Dateien erstellt, muss die Fontexpansion explizit aktiviert werden. Weder Letterspacing noch *automatische* Fontexpansion werden funktionieren, da die postprocessing Treiber (`dvips`, `dvipdfm` etc.) sowie die DVI-Betrachter nicht in der Lage sind, die Fonts on the fly zu generieren.

draft	true, false	false

final   Falls die `draft`-Option an das Paket geleitet wird, werden *alle mikro-typographischen Erweiterungen ausgeschaltet*, was zu unterschiedlichen Zeilen- und somit Seitenumbrüchen führen kann. Die `draft` und `final` Optionen können außerdem von den Optionen der Klasse geerbt werden; natürlich kann man diese mit den Optionen der Pakete überschreiben. Falls man z. B. die Klassenoption `draft` verwendet, um jegliche übervollen Boxen anzuzeigen, sollte man microtype mit der `final` Option laden.

verbose	true, false, errors, silent	false

Informationen bzgl. der Einstellungen an jeglichen Fonts werden in die Log-Datei geschrieben, sofern man die `verbose` Option aktiviert. Wenn microtype auf ein nicht-fatales Problem trifft (z. B. ein unbekanntes Zeichen in den Einstellungen oder nicht-existente Einstellungen), gibt es standardmäßig nur eine Warnung aus und versucht weiterzumachen. Lädt man das Paket mit `verbose=errors` werden alle Warnungen in Fehler verwandelt, sodass man sicher sein kann, dass kein Problem unerkannt bleibt. Falls man andererseits allen Warnungen nachgegangen ist und sich entscheidet, diese zu ignorieren, kann man microtype mittels `verbose=silent` zum Schweigen bringen.

babel	true, false	false

Das Paket mit der `babel` Option zu laden wird die Zeichensetzung der jeweiligen ausgewählten Sprache anpassen. Mehr Information hierzu in Abschnitt 6.

config	⟨*file name*⟩	microtype

Verschiedene Einstellungen für dieses Paket werden aus einer Hauptkonfigurationsdatei geladen, standardmäßig ist dies `microtype.cfg` (siehe Abschnitt 5.7). Sie können eine andere Konfigurationsdatei wählen, indem sie deren Name mittels *ohne die Dateiendung* angeben, z. B. `config=mycrotype`.

3.6 Nachträglicher Optionenwechsel

`\microtypesetup`   {⟨*key = value list*⟩}

Innerhalb der Präambel akzeptiert dieses Kommando alle oben genannten Paketoptionen (außer `config`). Im Dokumentkörper kann dieser Aufruf verwendet werden, um grundlegende Einstellungen der mikro-typographischen Erweiterungen zu verändern. Es akzeptiert dann alle Optionen aus Abschnitt 3.1: expansion, protrusion und activate, welche wiederum die Werte `true`, `false`, `compatibility` oder

nocompatibility erhalten können, `tracking`, `kerning` und `spacing` mit den zulässigen Werten `true` oder `false`. Den Namen eines Fonts an die Option zu leiten ist nicht erlaubt. Benutzt man dieses Kommando, so könnte man z. B. temporär Fontexpansion ausschalten, indem man wie folgt verfährt:

```
\microtypesetup{expansion=false}
```

4 Fonts für Mikrotypographie auswählen

Standardmäßig wird Character Protrusion auf alle Textschriftarten angewandt, die im Dokument genutzt werden und eine grundlegende Auswahl an Fonts wird der Fontexpansion unterzogen. Sie wollen vielleicht anpassen, welche Schriftarten den Vorzug bekommen sollten, mikro-typographisch behandelt zu werden. Dies kann geschehen, indem man 'Font Sets' deklariert und auswählt; diese Font Sets werden mittels Font Attributen spezifiziert, die zutreffen müssen.

`\DeclareMicrotypeSet`   [⟨*features*⟩] {⟨*set name*⟩} {⟨*set of fonts*⟩}

`\DeclareMicrotypeSet*`   Dieses Kommando deklariert ein neues Font Set, auf welches die mikro-typographischen Erweiterungen angewendet werden. Das optionale Argument kann eine mit Komma separierte Liste von Features enthalten, auf die das jeweilige Set beschränkt werden soll. Die mit Stern versehene Variante deklariert *und* aktiviert das Font Set gleichzeitig.

Das Font Set wird durch NFSS-Font-Attributen zugewiesene Werte definiert: Kodierung, Familie, Serie, Form und Größe (siehe auch LaTeX 2_ε font selection). Wir fangen mit einem Beispiel an. Dieses Paket definiert einen Font namens 'basictext' in der Hauptkonfigurationsdatei wie folgt:

```
\DeclareMicrotypeSet{basictext}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5},
     family   = {rm*,sf*},
     series   = {md*},
     size     = {normalsize,footnotesize,small,large}
   }
```

Wenn Sie nun

```
\UseMicrotypeSet[protrusion]{basictext}
```

in der Präambel des Dokuments aufrufen, werden nur Fonts mit den Kodierungen OT1, T1, T2A, LY1, OT4, QX oder T5 aus Roman oder Sans Serif Familien, normalen (oder 'medium') Serien und in per `\normalsize`, `\footnotesize`, `\small` oder `\large`aufgerufenen Größen hervorragen. Mathematische Schriftarten andererseits werden das nicht, da sie anderen Kodierungen angehören; gleiches gilt für fettgedruckte oder große Schriftarten etc.

Falls eine Attributsliste leer ist oder fehlt – wie das 'shape'-Attribut im obigen Beispiel – führt dies zu keiner Einschränkung, sprich, es ist äquivalent dazu *alle* möglichen Werte für dieses Attribut einzusetzen. Deswegen ist das vordefinierte Set 'alltext', welches wie folgt deklariert ist:

```
\DeclareMicrotypeSet{alltext}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
```

viel weniger restriktiv. Die einzige Bedingung lautet, dass die Kodierung stimmen muss.

Folgt einem Wert ein Asterisk (wie '`rm*`' und '`sf*`' im ersten Beispiel), kennzeichnet dieser keinen NFSS-Code, sondern wird in das \⟨*value*⟩`default` des Dokuments übersetzt, z. B. `\rmdefault`.[5] Ein einzelner Asterisk bedeutet \⟨*attribute*⟩`default`, z. B. `\encodingdefault`, gleichermaßen `\normalsize` für die Größenachse. Größen können entweder als Dimension angegeben werden ('`10`' oder '`10pt`') oder als Größenwahlkommando *ohne* den Backslash. Sie können sogar Bereiche (z. B. '`small-Large`') wählen; während die untere Grenze im Bereich enthalten ist, gilt dies nicht für die obere Grenze. Deshalb würde '`12-16`' beispielsweise 12 pt, 13.5 pt und 15.999 pt enthalten, aber nicht 16 pt. Es ist nicht erlaubt, die untere oder obere Grenze auszulassen ('`-10`', '`large-`').

Zusätzlich zu diesem Deklarationsschema können sie einzelne Schriften zu einem Set hinzufügen, indem Sie den '`font`'-Schlüssel nutzen, welcher die Aneinanderreihung aller Fontattribute, getrennt durch Slashes, erwartet, das heißt, '`font` = ⟨*encoding*⟩/⟨*family*⟩/⟨*series*⟩/⟨*shape*⟩/⟨*size*⟩'. Das erlaubt uns, Font Sets zu Sets hinzuzufügen, von denen sie sonst disjunkt sind. Falls Sie zum Beispiel wollen, dass die Roman Familie in allen Größen hervorragt, aber nur die normal große, möglicherweise kursive typewriter Font (im Kontrast zum z. B. kleinen Font), so könnten sie das Set so deklarieren:

```
\DeclareMicrotypeSet[protrusion]
   { myset }
   { encoding = T1,
     family   = rm*,
     font     = {T1/tt*/m/n/*,
                 T1/tt*/m/it/*} }
```

Wie Sie dem Beispiel entnehmen können, ist die Asterisk Notation auch für den `font`-Schlüssel zulässig. Ein einzelner Asterisk ist äquivalent zu '`*/*/*/*/*`', also dem normalen Font. Befehle zur Größenwahl sind auch möglich, allerdings sind keine Bereiche erlaubt.

Tabelle 2 listet die 9 vordefinierten Font Sets auf. Diese können ebenfalls aktiviert werden, indem ihr Name an die Optionen `protrusion`, `expansion`, `tracking`, `kerning` und `spacing` der Features geleitet wird, wenn das Paket geladen wird, z. B.:

```
\usepackage[protrusion=allmath,tracking=smallcaps]{microtype}
```

`\UseMicrotypeSet`   [⟨*features*⟩] {⟨*set name*⟩}

Dieses Kommando aktiviert ein vorher durch `\DeclareMicrotypeSet` deklariertes Font Set. Nutzt man das optionale Argument, so kann man begrenzen, wieviele

---

[5] Diese Übersetzungen geschehen zum `\AtBeginDocument`, was bedeutet, dass Änderungen an den Standardwerten innerhalb der Präambel auch berücksichtigt werden. Nur falls Font Standards `\AtBeginDocument` vom Benutzer selbst geändert werden, müssen Sie microtype nach diesen Änderungen laden.

Tabelle 2: Vordefinierte Font Sets

Set Name	Font attributes				
	Kodierung	Familie	Serie	Shape	Größe
all	∅	∅	∅	∅	∅
alltext (**allmath**)	Textkodierungen, TS1 (OML, OMS, U)	∅	∅	∅	∅
basictext (**basicmath**)	Textkodierungen (OML, OMS)	\rm*, \sf*	\md*	∅	\normalsize, \footnotesize, \small, \large
smallcaps	Textkodierungen	∅	∅	\sc*	∅
footnotesize	Textkodierungen, TS1	∅	∅	∅	-\small
scriptsize	Textkodierungen, TS1	∅	∅	∅	-\footnotesize
normalfont	\encoding*	\family*	\series*	\shape*	\normalsize

'Textkodierungen' = OT1, T1, T2A, LY1, OT4, QX, T5   '\...*' = '\...default'

Features auf das Set angewendet werden. Diese Anweisung hat nur einen Effekt, wenn das Features in den Paketoptionen aktiviert wurde.

\DeclareMicrotypeSetDefault [⟨*features*⟩] {⟨*set name*⟩}

Falls ein Feature aktiviert ist, aber kein Font Set explizit gewählt wurde, so werden die von diesem Befehl deklarierten Sets aktiviert. Standardmäßig wird das 'alltext' Font Set für Zeichenhervorhebungen und zusätzliches Kerning genutzt, das 'basictext' Set für Fontexpansion und Wortzwischenräume und das 'smallcaps' Set für Tracking.

Diese Anweisungen können nur in der Präambel oder der Hauptkonfigurationsdatei verwendet werden. Ihre Bandbreite gilt global für das Dokument. Nur ein Set pro Feature kann aktiviert werden.

5 Micro-Feinabstimmung

Jedes Zeichen bedarf eines bestimmten Betrags an Protrusion, Kerning oder Spacing. Es kann ebenso wünschenswert sein, die maximale Expansion bestimmter Zeichen zu begrenzen. Überdies, da jeder Font anders aussieht, müssen Einstellungen speziell auf eine Schriftart oder ein Set von Fonts zutreffen. Dieses Paket enthält flexible und einfache Methoden, um diese feineren Aspekte der Mikrotypographie anzupassen.

Alle Anweisungen zur Feinabstimmung folgen prinzipiell derselben Syntax: Sie benötigen alle 3 Argumente; das erste ist optional und kann zusätzliche Optionen enthalten; im zweiten Argument werden Font Sets spezifiziert, auf welche die Einstellungen angewendet werden sollen; das dritte Argument enthält die eigentlichen

Einstellungen.

Das Font Set, für welches die Einstellungen greifen sollen, wird mit der selben Syntax von ⟨*font axis*⟩ = ⟨*Werteliste*⟩ Paaren deklariert wie für das Kommando `\DeclareMicrotypeSet` (siehe Abschnitt 4). Der einzige Unterschied liegt darin, dass Werte mit anführendem Asterisk sofort übersetzt werden, nicht am Ende der Präambel.

Um die passenden Einstellungen für eine gegebene Schriftart zu finden, versucht das Paket alle Kombinationen von Fontkodierung, Familie, Serie, Form und Größe, mit abnehmendem Stellenwert in dieser Reihenfolge. Existieren zum Beispiel sowohl Einstellungen für die momentanen Familien (z. B. `T1/cmr///`) als auch Einstellungen für kursive Fonts mit normalem Gewicht (`T1//m/it/`), so würden die Einstellungen für die `cmr` Familie benutzt.[6] Die Kodierung muss immer stimmen.

5.1 Randausgleich (Character protrusion)

`\SetProtrusion`   [⟨*options*⟩] {⟨*set of fonts*⟩} {⟨*protrusion settings*⟩}

Benutzt man dieses Kommando, so kann man die Faktoren für Protrusion (= optischer Randausgleich) jedes einzelnen Zeichens einer Schriftart oder eines Font Sets verändern. Ein sehr unvollständiges Beispiel wäre das folgende:

```
\SetProtrusion
   { encoding = T1,
     family   = cmr }
   { A               = {50,50},
     \textquoteleft = {700, } }
```

Dies würde darin resultieren, dass der Buchstabe 'A' um 5% seiner Breite auf beiden Seiten hervorragt; das linke Anführungszeichen würde hier um 70% seiner Breite zum linken Rand hin hervorragen. Angewandt wird das Beispiel auf alle Font-Formen, Serien und Größen der T1 kodierten Computer Modern Roman Familie.

Die Protrusionseinstellungen   bestehen aus ⟨*Zeichen*⟩ = ⟨*Protrusionsfaktoren*⟩ Paaren.

Die Zeichen können entweder einzeln als solche angegeben werden ('`A`'), als ein Symbolkommando ('`\textquoteleft`'), oder als Slotnummer: 3 Stellen für dezimale Notation, angeführt von `"` für hexadezimal, mit `'` für oktal (z. B. die 'fl' Ligatur in T1-Kodierung: `029`, `"1D`, `'35`). 8-bit (und sogar UTF-8) Zeichen können direkt oder in LATEX's traditioneller 7-bit Notation eingegeben werden: sowohl `\"A` als auch Ä sind zulässig, vorausgesetzt, dass das Zeichen sowohl in der Input- als auch der Fontkodierung deklariert wird. Sie haben ebenfalls die Möglichkeit, Zeichenlisten anzugeben, für welche die Einstellungen übernommen werden sollen (siehe Abschnitt 5.6).

Die Protrusionsfaktoren weisen einem Zeichen den Betrag zu, um den sie zum linken Rand hin (erster Wert) bzw. rechten Rand hin (zweiter Wert) hervorragen sollen. Standardmäßig sind die Werte relativ zur Breite des Zeichens zu verstehen, sodass ein Wert von 1000 bedeutet, dass das Zeichen vollständig zwischen die Ränder gelegt werden sollte, während es z. B. bei einem Wert von 50 um 5% seiner

---

6   Für alle Interessierten stellt Tabelle 4 auf Seite 87 die exakte Reihenfolge dar.

Breite hervorragen würde. Negative Werte sind erlaubt, ebenso Zahlen größer als 1000 (effektiv jedoch nicht mehr als 1 em der Schriftart). Sie können jede der beiden Zahlen auslassen, wenn Sie nicht wollen, dass das jeweilige Zeichen zu dieser Seite hervorragen sollte. Das abtrennende Komma jedoch dürfen Sie nicht auslassen.

Optionen:

`name` Sie können den Protrusionseinstellungen einen Namen zuweisen, sodass Sie in der Lage sind, sie aus einer anderen Liste zu laden.

`load` Sie können eine andere Liste laden (vorausgesetzt, dass Sie dieser zuvor einen Namen zugewiesen haben) bevor die aktuelle Liste geladen wird, wodurch die Fonts die Werte der geladenen Liste erben.

Dadurch kann die Konfiguration beträchtlich erleichtert werden. Sie können zum Beispiel eine Standardliste für eine Schriftart erstellen; Einstellungen für andere Formen oder Serien können diese Einstellungen laden, sie erweitern und überschreiben (da der jeweils letzte Wert Vorrang hat). Fonteinstellungen werden rekursiv geladen. Die folgenden Optionen werden alle geladenen Listen beeinflussen:

`factor` Diese Option kann genutzt werden, um alle Protrusionsfaktoren der Liste zu beeinflussen, was alle globalen `factor` Einstellungen überschreibt (siehe Abschnitt 3.2). Wollen sie zum Beispiel, dass Fonts mit größeren Abmaßen weniger hervorragen, könnten Sie die normalen Listen laden, allerdings könnten Sie kleinere Faktoren darauf anwenden:

```
\SetProtrusion
   [ factor   = 700
     load     = cmr-T1 ]
   { encoding = T1,
     family   = cmr,
     size     = large- }
   { }
```

`unit` Standardmäßig sind die Protrusionsfaktoren relativ zur Breite des jeweiligen Zeichens. Die `unit`-Option kann genutzt werden, um dieses Verhalten zu überschreiben, sodass microtype diese Werte als Tausendstel der angegeben Werte deutet. Gibt man z. B. die Anweisung 'unit=1em' hätte dies den Effekt, dass ein Wert von 50 nun darin resultieren würde, dass ein ein Zeichen um 5% eines em des Fonts hervorragen würde (und somit die interne Vermessung von pdfTEX's `\lpcode` und `\rpcode` primitives simuliert). Das normale Verhalten kann mittels `unit=character` wiederhergestellt werden.[7]

`preset` Setzt die Protrusions Codes aller Zeichen auf die angegebenen Werte (={⟨*left*⟩,⟨*right*⟩}), u. U. skaliert mit einem `factor`. Eine `unit` Einstellung wird nur berücksichtigt, wenn sie nicht =character ist.

`inputenc` Wählt eine Inputkodierung aus, die auf die Liste angewendet werden soll, unabhängig von der Eingabekodierung des Dokuments selbst. Sie können jede

---

7 Die `unit`-Option kann auch global an das Paket geleitet werden (siehe Abschnitt 3.2). Allerdings werden alle vorliegenden Einstellungen unter der Annahme, dass die Werte relativ zur Zeichenbreite sind, erstellt. Deshalb sollten Sie es nur ändern wenn Sie sich sicher sind, dass die Standardeinstellungen im Dokument nicht verwendet werden.

Kodierung angeben, die per `inputenc` geladen werden könnte, also u. a. `ansinew`, `koi8-r`, `utf8`.

`context` Die Bandbreite der Liste kann auf einen gewissen Kontext limitiert werden. Schauen Sie in Abschnitt 6 für ein Anwendungsbeispiel nach.

5.2 Fontexpansion

`\SetExpansion` [⟨*options*⟩] {⟨*set of fonts*⟩} {⟨*expansion settings*⟩}

Standardmäßig können alle Zeichen eines Fonts um denselben Betrag gedehnt oder geschrumpft werden. Man kann aber auch die Expansionslimits für bestimmte einzelne Zeichen festlegen, wenn diese sensibler auf Deformation reagieren. Das ist der Zweck des `\SetExpansion`-Kommandos. Beachten Sie, dass dies nur einen Effekt hat, wenn das Paket mit der Option `selected` geladen wurde (siehe Abschnitt 3.3). Ansonsten werden die Expansionseinstellungen ignoriert – anders als die Optionen im optionalen ersten Argument, welche trotzdem ausgewertet werden.

Falls das Paket mit der `selected`-Option geladen wurde und Einstellungen für einen Font nicht existieren, wird Fontexpansion gar nicht auf diesen Font angewendet. Sollte die außergewöhnliche Situation aufkommen, dass sie die ausgewählte Expansion generell verwenden wollen, aber dass alle Zeichen eines bestimmten Font (Sets) um denselben Betrag geschrumpft oder expandiert werden sollten, so müssen sie eine leere Liste für diese Fonts erstellen.

Die Expansionseinstellungen bestehen aus Paaren der Form ⟨*character*⟩ = ⟨*expansion factor*⟩. Sie können eine Zahl für jedes Zeichen festlegen, die den Betrag bestimmt, um den ein Zeichen expandiert werden kann. Die Zahlen bezeichnen Tausendstel der vollen Expansion. Setzt man zum Beispiel den Expansionsfaktor des Zeichens 'O' auf 500, wird dieses nur um die Hälfte des Betrages geschrumpft oder expandiert, um den die anderen Zeichen expandiert werden. Während der Standardwert für Character Protrusion auf 0 gesetzt ist – sofern Sie keine Zeichen angegeben haben, wird auch keines hervorspringen – ist der Standardwert für Expansion 1000, was bedeutet, dass alle Zeichen um denselben Betrag expandiert werden.

Optionen:

`name`, `load`, `preset`, `inputenc`, `context` Analog zu `\SetProtrusion` kann dieses optionale Argument genutzt werden, um einer Liste einen Namen zuzuweisen, eine andere Liste zu laden, alle Expansionsfaktoren voreinzustellen, die Eingabekodierung festzulegen oder den Kontext der Liste zu bestimmen (Expansionskontexte sind nur möglich mit pdfTeX Version 1.40.4 oder neuer).

`auto`, `stretch`, `shrink`, `step` Diese Schlüssel können genutzt werden, um die globalen Einstellungen der Paketoptionen zu überschreiben (siehe Abschnitt 3.3). Wenn Sie keine der Optionen `stretch`, `shrink` und `step` angeben, wird ihr jeweiliger globaler Wert genutzt (es findet also keine Berechnung statt).

Ein praktisches Beispiel: Nehmen wir an, wir haben einen Paragraphen, der eine Witwe enthält, die man einfach hätte verhindern können, indem man den Font ein wenig mehr geschrumpft hätte. In Verbindung mit der `context` Option (siehe Abschnitt 6 für weitere Details) könnte man mit diesem speziellen Paragraphen mehr Expansion erlauben:

```
\SetExpansion
   [ context = sloppy,
     stretch = 30,
     shrink  = 60,
     step    =  5  ]
   { encoding = {OT1,T1,TS1} }
   { }
% ... END PREAMBLE
{\microtypecontext{expansion=sloppy}%
 Dieser Paragraph enthält eine `unnötige' Witwe.}
```

Diese Methode, Kontext anzuwenden, um vorübergehend verschiedene Expansionsparameter zu setzen, funktioniert nur mit pdfTEX 1.40.4 oder neuer (für ältere Versionen wird ein schmutziger Trick in Abschnitt 14.2 auf Seite 59 beschrieben). Bedenken Sie auch, dass pdfTEX die Nutzung von Fonts mit verschiedenen Expansionslimits oder -stufen (sogar für verschiedene Fonts) innerhalb eines Paragraphs verbietet, weswegen der saloppe Kontext benutzt werden muss, um Paragraphen zu komplettieren.

factor   Diese Option stelle eine andere Methode bereit, um Expansionseinstellungen für bestimmte Fonts zu verändern, arbeitet aber um die eben beschriebenen Restriktionen herum. Die factor Option beeinflusst die Expansionsfaktoren aller Zeichen (in Kontrast zur allgemeinen Streckbarkeit) des Fonts. Wollten sie z. B. die kursive Form weniger expandieren lassen, würden sie vereinbaren:

```
\SetExpansion
   [ factor   = 500 ]
   { encoding = *,
     shape    = it }
   { }
```

Die factor-Option kann nur genutzt werden, um die Streckbarkeit der Zeichen zu *verringern*, weshalb sie nur Werte kleiner als 1000 erhalten darf. Außerdem kann sie nur für einzelne Fonts oder Font Sets genutzt werden; die Option global in den Paketoptionen zu setzen, macht wenig Sinn – dafür nutzt man die Optionen stretch und shrink des Pakets.

5.3   „Sperren" (Tracking)

\SetTracking   [⟨*options*⟩] {⟨*set of fonts*⟩} {⟨*tracking amount*⟩}

Eine bedeutende typographische Technik – welche lange in TEX fehlte – ist die Anpassung des Trackings, sprich die uniforme Addition oder Subtraktion von Sperrsatz zu/von allen Zeichen eines Fonts. Zum Beispiel ist es ein normal üblicher, typographischer Brauch, Kapitälchen (wie in diesem Dokument) ein wenig zu sperren. Die Lesbarkeit kann auch verbessert werden, indem man das Tracking von kleinerer Typen minimal erhöht und das größerer Typen minimal erhöht.[8] Das \SetTracking-Kommando erlaubt, die Menge des Trackings für verschiedene Fonts

---

8   Für Fonts mit vollem Funktionsumfang wie Computer Modern ist das für gewöhnlich nicht nötig, da sie in optischen Größen kommen und das Tracking der Kapitälchen bereits angepasst ist.

oder Font Sets zu setzen. Es wird vom `\textls` Kommando ausgewertet, welches genutzt werden kann, um kleinere Textpassagen zu sperren (siehe Abschnitt 7).

Die Menge des Tracking wird festgelegt in Tausendsteln von 1 em (oder der gegebenen Einheit); negative Werte sind ebenfalls erlaubt.

Optionen:

`name`, `unit`, `context`   Diese Optionen dienen derselben Funktionalität wie bei den vorigen Konfigurationskommandos. Die Einheit kann jegliche Dimension annehmen, der Standard beträgt 1 em.

`spacing`   Werden die *Zeichen*zwischenräume verändert, müssen auch die *Wort*zwischenräume vielleicht angepasst werden. Diese Option erwartet drei Zahlen für Wortzwischen-raum, bzw. Streckung und Schrumpfung, welche in Tausendsteln eines 1 em angege-ben werden (oder in der aktuellen Einheit (`unit`)). Folgt einem Wert ein Asterisk, so zeigt es an, dass Tausendstel der jeweiligen Fontdimension zu dieser hinzuaddiert werden. Zum Beispiel werden durch

```
\SetTracking[ spacing = {25*,166, } ]{ encoding = *, shape = sc }{ 25 }
```

die Wortzwischenräume um 2.5% erhöht, die Streckung wird auf einen Betrag von 0.166 em gesetzt, an den Schrumpf-Einstellungen wird nichts geändert. Falls sie die `spacing`-Option nicht angeben, wird der Wortzwischenraum über die aktuelle Größe der Sperrsatzabstände skaliert (wie in obigem Beispiel), wobei stretch und shrink unberührt bleiben.

`outer spacing`   Falls ein Wortzwischenraum direkt auf Sperrsatz folgt (oder umgekehrt), wird er standardmäßig gleich den Zwischenräumen im gesperrten Text behandelt. Mit dieser Option, welche dieselben Werte wie `spacing` akzeptiert, kann sie unabhängig eingestellt werden.

`outer kerning`   Falls andererseits kein Wortzwischenraum folgt oder vorgeht, können Sie immer noch die ersten und letzten Buchstaben von angrenzenden Buchstaben abtrennen. Diese Option erwartet durch Komma abgetrennte Kerning Beträge für die linke und rechte Seite, in Tausendsteln eines 1 em (oder der aktuellen `unit`). Folgt einem Wert ein Asterisk, bezeichnet dies Tausendstel der aktuellen Abstandsgröße des Sperrsatzes. Ein einzelner Asterisk bedeutet '`500*`'; das ist ebenso der Standard, sprich die Summe des äußeren Kerns beträgt standardmäßig soviel wie die aktuelle Größe der Abstände des Sperrsatzes. Um Kerning auf beiden Seiten zu entfernen, verwendet man '`outer kerning={0,0}`'.

`no ligatures`   Soweit pdfTEX betroffen ist, werden Ligaturen in gesperrten Fonts wie gewöhnlich konstruiert, was angebracht sein kann, wenn das Tracking um einen kleinen Betrag geändert wird. Für größere Abstände andererseits hätte der normale Abstand zwischen Zeichen innerhalb Ligaturen enttäuchende E f f e k t e. Dieser Schlüssel erwartet eine per Komma abgetrennte Liste von Zeichen, für welche die Ligaturen abgeschaltet werden sollen; nur das Zeichen, mit welchem die Ligatur beginnt, muss angegeben werden. Wird der Schlüssel ohne Wert angegeben, so werden *alle* Ligaturen des Fonts abgeschaltet. Das wird jedoch nicht empfohlen, da es ebenso mit sich bringt, dass Kerning ausgeschaltet wird.[9] Die Standardeinstel-

---

9   Die untrennbare Verbindung von Ligaturen und Kernen ist eine Begrenzung von TEX die nicht vor der Einführung luaTEX beseitigt wird.

lungen stellen Ligaturen nur für das Zeichen 'f' ab, sprich, 'ff', 'fi', ffi, etc.[10] In außergewöhnlichen Situationen können Sie manuell Ligaturen aufbrechen, indem sie '{\kern0pt}' bzw. babels "| Kürzel verwenden, oder aber die Ligaturen schützen, indem Sie sie in \lslig einschließen (siehe Abschnitt 7).

> *[Die originale Dokumentation[11] enthält ein Bild, welches all diese Optionen illustriert.]*

Nehmen wir zum Beispiel an, Sie wollen alle kleinen Kapitälchen um $50/1000$ em sperren, Fonts kleiner \small um 0.02 em, und das Tracking großer Typen um 0.02 em erhöhen, so können Sie das mit folgenden Einstellungen erreichen:

```
\usepackage[tracking=true]{microtype}
\DeclareMicrotypeSet*[tracking]{my}
   { encoding = *,
     size     = {-small,Large-},
     font     = */*/*/sc/* }
\SetTracking[ no ligatures = f ]{ encoding = *, shape = sc}{ 50 }
\SetTracking{ encoding = *, size = -small }{  20 }
\SetTracking{ encoding = *, size = Large- }{ -20 }
```

Gesperrte Schriftarten, für welche keine Einstellungen vorliegen, werden um den den Standard von 0.1 em gesperrt (einstellbar mit der Paketoption letterspace, siehe Abschnitt 3.5). Setzen wir voraus, ihr Herausgeber will, dass Sie ihre 1000 Seiten auf ein Meisterwerk von einer handvoll Seiten zusammenkürzen, dann könnten Sie (mit gekreuzten Fingern) microtype laden:

```
\usepackage[tracking=alltext,letterspace=-40]{microtype}
```

5.4 Zusätzliches Kerning

\SetExtraKerning [⟨*options*⟩] {⟨*set of fonts*⟩} {⟨*kerning settings*⟩}

Mit diesem Befehl können sie das zusätzliche Kerning fein abstimmen. Im Vergleich zu herkömmlichem Kerning, welches immer mit einem *Paar* von Zeichen assoziiert wird, und Tracking, welches den Zwischenraum zwischen *allen* Zeichen eines Fonts angibt, bezieht sich zusätzliches Kerning nur auf einzelne Zeichen, soll heißen, immer wenn ein bestimmtes Zeichen im Text auftaucht, wird das angegebene Kerning eingefügt, unabhängig davon, welche Zeichen vorgehen oder folgen.

Ich sollte nicht versäumen zu sagen, dass es eine Limitierung zusätzlichen Kernings gibt: Wörter, die solch einem Kern *direkt folgen* (nicht von einem Leerzeichen getrennt), werden nicht mit Bindestrich versehen, solang man die Trennpunkte nicht selbst einstellt, z. B. für Kerning nach dem Apostroph 'l'apos\-trophe'. Diese Restriktion von pdfTEX wird hoffentlich bald abgeschafft.

Die Einstellungen des Kerning   Sie werden als Paare der Form ⟨*Zeichen*⟩ = ⟨*Kerning Werte*⟩ angegeben, wobei letzteres 2 Werte enthält: das Kerning, welches vor und nach dem betreffenden Zeichen angefügt wird. Einmal mehr können Werte ausgelassen werden, nicht aber das Komma.

---

10   Mit pdfTEX Versions vor 1.40.4 sind *alle* Ligaturen und somit jegliches Kerning abgeschaltet. Es wird deshalb empfohlen, wenigstens Version 1.40.4 zu nutzen.
11   Available from CTAN at /macros/latex/contrib/microtype/microtype.pdf.

Optionen:

`name`, `load`, `factor`, `preset`, `inputenc`   Diese Optionen dienen der gleichen Funktionalität wie in den vorigen Konfigurationsbefehlen.

`unit`   Zulässige Werte sind: `space`, `character` und eine ⟨*Dimension*⟩. Standardmäßig bezeichnen die Werte ein Tausendstel von 1 em.

`context`   Wenn es um Kerning Einstellungen geht, ist diese Option besonders nützlich, da sie erlaubt, Einstellungen abhängig von der Sprache anzuwenden.

Zum Beispiel kann man die folgenden Einstellungen, angedacht für Dokumente in Französisch, in der Hauptkonfigurationsdatei finden:

```
\SetExtraKerning
   [ name      = french-default,
     context   = french,
     unit      = space   ]
   { encoding = {OT1,T1,LY1} }
   {
     :  = {1000,}, % = \fontdimen2
     ;  = {500, }, % ~ \thinspace
     !  = {500, },
     ?  = {500, }
   }
```

Was ist das Ergebnis dieser Einstellungen ? Wenn sie aktiv sind wie im vorliegenden Paragraphen, wird ein schmales Leerzeichen vor jedem Fragezeichen, Ausrufezeichen und Semikolon eingefügt ; ein normales Leerzeichen vorm Doppelpunkt. In Abschnitt 6 wird beschrieben, wie diese Einstellungen aktiviert werden. Dieser Paragraph wurde wie folgt eingegeben :

```
\begin{microtypecontext}{kerning=french}
  Was ist das Ergebnis dieser Einstellungen? Wenn sie aktiv sind wie
  im vorliegenden Paragraphen, wird ein schmales Leerzeichen vor
  jedem Fragezeichen, Ausrufezeichen und Semikolon eingefügt; ein
  normales Leerzeichen vorm Doppelpunkt.
  In Abschnitt~\ref{sec:context} wird beschrieben, wie diese Einstellungen
  aktiviert werden. Dieser Paragraph wurde wie folgt eingegeben:
\end{microtypecontext}
```

5.5   Wortzwischenräume

`\SetExtraSpacing`   [⟨*Optionen*⟩] {⟨*Font Set*⟩} {⟨*Spacing Einstellungen*⟩}

Dieser Befehl erlaubt Ihnen, die Wortzwischenräume fein abzustimmen (auch bekannt als glue). Eine vorausgehende Bemerkung darüber, was ein 'Zwischenraum' ist, könnte angebracht sein: Zwischen zwei Wörtern fügt TEX sogenannten glue ein, welcher von drei Parametern charakterisiert wird – der normale Abstand zwischen zwei Wörtern, der maximale Abstand, der dem angefügt werden darf, und das Maximum welches davon subtrahiert werden kann. Die letzteren beiden Parameter wirken, wann immer TEX versucht, einen Paragraphen in Zeilen zu zerlegen und dabei scheitert; es kann die Räume zwischen Wörtern strecken oder schrumpfen. Diese drei Parameter sind spezifisch für jeden Font.

Obendrein beherrscht TeX das Konzept von 'space factors'. Sie können genutzt werden, um die Räume zwischen bestimmten Zeichen, besonders den Satzzeichen zu vergrößern. Falls pdfTeX zusätzliche Zwischenräume in Betrieb sind, werden space factors ignoriert, da es als Erweiterung zu space factors mit viel präziserer Kontrolle betrachtet werden kann.

Einstellungen an den Zwischenräumen werden als Paare der Form ⟨*Zeichen*⟩ = ⟨*Zwischenraum Faktoren*⟩ deklariert, wobei letztere aus drei Zahlen bestehen: zuerst der zusätzliche Kern, der nach diesem Zeichen angefügt wird, wenn es vor einem Wortzwischenraum vorkommt, 2. der zusätzliche Streckungsbetrag und zuletzt der zusätzliche Schrumpfbetrag. Alle Werte können auch negativ sein, wodurch die Dimensionen verkleinert werden. Nicht alle Werte müssen angegeben werden, die Einstellungen müssen jedoch zwei abtrennende Kommas beinhalten.

Optionen:

`name`, `load`, `factor`, `preset`, `inputenc`, `context` Diese Optionen dienen der gleichen Funktionalität wie in den vorigen Konfigurationsbefehlen.

`unit` Sie können die Einheit wählen, in der die angegebenen Zahlen gemessen werden. Mögliche Werte sind: `character`, eine ⟨*Dimension*⟩ und zusätzlich `space`. Letztere Einstellung misst die Werte in Tausendsteln der jeweiligen Dimension, die vom Font eingestellt ist. Standardmäßig wird die Einheit in Zwischenraumdimensionen gemessen. Mit folgenden (unsinnigen) Einstellungen z. B.

```
\SetExtraSpacing
   [ unit = space  ] % default
   { font = */*/*/* }
   {
         .  = {1000,1000,1000},
   }
```

würde der Raum nach einem Punkt verdoppelt (streng genommen: $2 \times$ `\fontdimen` 2), ebenso die Schrumpfbeträge der Wortzwischenräume (`\fontdimen` 3 und 4). Umgekehrt würde man, so man alle drei Werte auf −1000 setzt, das Leerzeichen nach dem betreffenden Zeichen komplett aufheben.

5.6 Zeichenvererbung

`\DeclareCharacterInheritance` [⟨*Features*⟩] {⟨*Font Set*⟩} {⟨*Vererbungslisten*⟩}

In den meisten Fällen sollten akzentuierte Zeichen die Einstellungen ihres Ursprungszeichens erben. Zum Beispiel sollten vermutlich alle der Zeichen À, Á, Â, Ã, Ä, Å und Ă um denselben (absoluten) Betrag hervorragen wie der Buchstabe A. Nutzt man den Befehl `\DeclareCharacterInheritance`, so kann man Klassen von Zeichen benennen, wodurch man nur das betreffende Grundzeichen einstellen muss. Mit dem zusätzlichen Argument, welches eine per Komma abgetrennte Liste von Features enthalten kann, können Sie die Reichweite der Liste beschränken. Zusätzlich dazu akzeptiert es den `inputenc` Schlüssel, um die Eingabekodierung für diese Liste einzustellen. Das Font Set kann wie üblich eingestellt werden. Die einzige Ausnahme hierbei ist, dass exakt eine Kodierung gewählt werden muss. Die Vererbungslisten werden als Paare der Form ⟨*base character*⟩ = ⟨*Liste der erbenden*

Zeichen⟩ deklariert. Solange Sie keine abweichende Kodierung oder einen sehr eigenartig geformten Font verwenden, sollte es keinen Grund geben, die standardmäßigen Vererbungseinstellungen zu ändern.

In der Hauptkonfigurationsdatei `microtype.cfg` und den anderen font-spezifischen Konfigurationsdateien können Sie Beispiele zu all diesen Befehlen finden.

5.7 Konfigurationsdateien

Die Standardkonfiguration beinhaltet Vererbungseinstellungen, Deklarationen von Font Sets und alias Fonts sowie generische Hervorragung, Expansion, Spacing und Kerning Einstellungen, und wird aus der Datei `microtype.cfg` geladen. Sie können diese Datei mit eigenen Einstellungen erweitern (oder eine andere Konfigurationsdatei mit der 'config' Option laden, siehe Abschnitt 3.5).

Wenn Sie den Weg einschlagen, neue Einstellungen für eine Fontfamilie zu treffen, sollten Sie diese in eine separate Datei stecken, deren Name 'mt-⟨*Fontfamilie*⟩.cfg' (z. B. 'mt-cmr.cfg') lauten muss, und welche alle Befehle enthalten muss, die in diesem Abschnitt 5 beschrieben. Diese Dateien werden automatisch geladen, wenn sie die betreffenden Fonts tatsächlich nutzen. Das Paket kommt mit mit Konfigurationsdateien für eine Reihe von Fontfamilien; Tabelle 3 enthält sie alle.

\DeclareMicrotypeVariants {⟨*list of suffixes*⟩}

\DeclareMicrotypeVariants* Auf seiner Suche nach einer Konfigurationsdatei versucht das Paket auch, Suffixe oder einen oder mehr Buchstaben von einem Fontnamen zu entfernen, welche auf eine 'Variante' des Basisfonts hindeuten könnten (vergleiche Karl Berrys Fontname). Das erlaubt beispielsweise Einstellungen für z. B. die Fonts `padx` (Experten Set), `padj` (altertümliche Numerale) und `pad` (einfach) in der selben Datei `mt-pad.cfg` vorzunehmen. Der Befehl erwartet eine durch Komma abgetrennte Liste von Suffixen für Varianten. Die Version mit Stern hängt die Suffixe der existierenden Liste an. Die Standard-Deklaration in `microtype.cfg` lautet:

```
\DeclareMicrotypeVariants{x,j,w,a,d,0,1}
```

\DeclareMicrotypeAlias {⟨*Fontname*⟩} {⟨*alias Font*⟩}

Dieser Befehl kann für Fonts genutzt werden, die sich sehr ähnlich oder gar gleich sind (falls sie sich zum Beispiel nicht an das Berry Namensgebungsschema beim Installieren der Fonts hielten). Ein Beispiel wären die Latin Modern Fonts, welche von Computer Modern abgeleitet sind, sodass es nicht notwendig ist, neue Einstellungen für diese vorzunehmen – man könnte sagen:

```
\DeclareMicrotypeAlias{lmr}{cmr}
```

Dadurch probiert das Paket – wann immer der Font `lmr` vorkommt und keine Einstellungen dafür gefunden werden können – auch den Font `cmr` aus. Tatsächlich werden Sie eben diese Zeile (mit einigen anderen) in der Standardkonfigurationsdatei vorfinden.

\LoadMicrotypeFile {⟨*Fontname*⟩}

In wenigen Fällen kann es nötig sein, eine Fontkonfigurationsdatei manuell zu laden (z. B. aus einer anderen Konfigurationsdatei heraus), oder in der Lage zu sein, in

Tabelle 3: Fonts mit zugeschnitten Hervorragungseinstellungen

Fontfamilie (NFSS code)	Features	
	Kodierungen	Formen
Generic	OT1, T1, T2A, LY1, QX, (TS1)[a]	n, (it, sl, sc)[a]
Computer Modern Roman (`cmr`)[b]	OT1, OT4, T1, T2A, T5, LY1, TS1	n, it, sl, sc
Bitstream Charter (`bch`)[c]	OT1, T1, T5, LY1, TS1	n, it, (sl)[d], sc
Adobe Garamond		
(`pad`, `padx`, `padj`)	OT1, T1, LY1, TS1	n, it, (sl)[d], sc
URW Garamond (`ugm`)[e]	OT1, T1, TS1	n, it
Bitstream Letter Gothic (`blg`)[f]	OT1, T1, TS1	n, it
Adobe Minion (`pmnx`, `pmnj`)	OT1, T1, T2A, LY1, TS1	n, it, (sl)[d], sc, si
Palatino (`ppl`, `pplx`, `pplj`)[g]	OT1, OT4, T1, LY1, (TS1)[a]	n, it, (sl)[d], sc
Times (`ptm`, `ptmx`, `ptmj`)[h]	OT1, OT4, T1, LY1, QX, (TS1)[a]	n, it, (sl)[d], sc
Computer Modern math		
(`cmsy`, `cmm`)	OML/OMS	n/it
AMS symbols (`msa`, `msb`)	U	n
Euler (`eur`, `eus`, `euf`)[i]	U	n
Euro symbols		
(Adobe, ITC ,marvosym)	U/OT1	n, it

a	Unvollständig
b	Alias: Latin Modern (`lmr`), ae (`aer`), zefonts (`zer`), eco (`cmor`), hfoldsty (`hfor`)
c	Alias: mathdesign/Charter (`mdbch`), MicroPress's chmath (`chr`)
d	Einstellungen vererbt von kursiver Form
e	Alias: mathdesign/URW Garamond (`mdugm`)
f	Alias: ulgothic (`ulg`)
g	Alias: pxfonts (`pxr`), qfonts/QuasiPalatino, TEX Gyre Pagella (`qpl`), FPL Neu (`fp9x`, `fp9j`)
h	Alias: txfonts (`txr`), qfonts/QuasiTimes, TEX Gyre Termes (`qtm`)
i	Alias: eulervm (`zeur`, `zeus`)

einer Datei festgelegte Einstellungen zu erweitern, die sonst nicht automatisch oder zu spät geladen würden.[12] Dieser Befehle lädt die Datei 'mt-⟨*Fontname*⟩.cfg'.

6 Kontext-sensitive Einstellungen

Das microtype -Paket erlaubt zudem verschiedene mikro-typografische Einstellungen für die Schriftarten zu verwenden, die vom Zusammenhang abhängig sind. Dies eröffnet unendlich viele Möglichkeiten der Feinabstimmungen für das Aussehen des Dokuments.

\microtypecontext {⟨*context assignments*⟩}

Dieser Befehl kann überall im Dokument (auch in der Präambel) verwendet werden, um den mikro-typografischen Zusammenhang in der aktuellen Gruppe zu

---

12 Autoren von Fontpaketen sollten vielleicht einen Blick auf den hook \Microtype@Hook werfen, beschrieben im Implementationsteil, Abschnitt 14.4.3.

ändern. Jeder Funktion (protrusion, expansion, tracking, spacing und kerning) kann ein Kontext zugewiesen werden. Folglich können nur Einstellungen mit dem entsprechenden 'Kontext'-Schlüsselwort verwendet werden.

\begin{microtypecontext} {⟨*context assignments*⟩}
\end{microtypecontext} Wieviele andere LaTeX -Befehle ist der Befehl in Form einer Umgebung verfügbar.

\textmicrotypecontext {⟨*context assignments*⟩} {⟨*general text*⟩}

Als weitere Möglichkeit ändert der Befehl \textmicrotypecontext den Kontext für den Text, der im zweiten Argument übergeben wird.

Angenommen, Sie möchten eine größere Menge von Fussnotezeichen im Text erhalten, dann können die Einstellungen der Zahlen so definiert werden,

```
\SetProtrusion
   [ context = footnote ]
   { font    = */*/*/*/scriptsize } % adapt if necessary
   { 1 = { ,650}, 2 = { ,400}, 3 = { ,400}, 4 = { ,400}, 5 = { ,400},
     6 = { ,400}, 7 = { ,500}, 8 = { ,400}, 9 = { ,400}, 0 = { ,400} }
```

um damit den Kontext im Fussnoten-Befehl zu ändern. Dieser Befehl unterscheidet sich zwischen den verschiedenen Klassen. Für die Basisklassen, z. B. article, wäre es:

```
\newcommand*\new@makefnmark{\hbox{\@textsuperscript{\normalfont
   \microtypecontext{protrusion=footnote}\@thefnmark}}}
\renewcommand*\@footnotemark{%
   \leavevmode \ifhmode\edef\x@sf{\the\spacefactor}\nobreak\fi
   \new@makefnmark \ifhmode\spacefactor\x@sf\fi \relax}
```

Für die memoir Klasse, müssten Sie zusätzlich die automatische Erkennung mehrerer Fussnoten deaktivieren, welches ein Hervorheben des Fussnotentextes verhindert.

```
\renewcommand*\@makefnmark{\hbox{\@textsuperscript{\normalfont
   \microtypecontext{protrusion=footnote}\@thefnmark}}}
\let\m@mmf@prepare\relax
\let\m@mmf@check\relax
```

Eine weitere Möglichkeit wäre es, den Kontext für eine sprachabhängige Einstellung zu verwenden. Zum Beispiel, wenn Sie einen Text auf Französisch schreiben, können Sie

```
\microtypecontext{kerning=french}
```

in der Präambel hinzufügen. Das hätte den Effekt, dass die Abstandseinstellungen des französischen Kontextes, für das Dokument übernommen würden. Sollten Teile des Dokuments in Englisch sein, könnten Sie schreiben:

```
\textmicrotypecontext{kerning=}{English text!}
```

um den Kontext zurückzusetzen, so dass die Interpunktion in diesen Teilen kein extra Zeichenabstand erhalten.

Anstatt diese Befehle manuell zu ihrem Dokument hinzuzufügen, können Sie microtype auch mit der Babel-Funktion laden (siehe Abteilung 3.5). Die aktuelle Sprache wird dann automatisch erkannt und der Kontext dementsprechend angepasst.

\DeclareMicrotypeBabelHook {⟨*list of babel languages*⟩} {⟨*context list*⟩}

Natürlich kennt microtype nicht jede typografische Besonderheit jeder einzelnen Sprache. Dieser Befehl ist ein Mittel, um zu lehren, wie man den Kontext anpassen kann, wenn eine bestimmte Sprache ausgewählt ist. Die Hauptkonfigurationsdatei enthält unter anderen die folgende Deklaration:

```
\DeclareMicrotypeBabelHook
    {french,francais,acadian,canadien}
    {kerning=french, spacing=}
```

Folglich, wann immer Sie auf französische Sprache umschalten, wird die Abstandseinstellung im Kontext auf 'französisch' geändert und der Abstandskontext zurückgesetzt. Dieser Effekt greift nur dann, wenn das Paket mit der babel Option geladen wurde. Derzeit unterstützt microtype französische, türkische und englische Buchstabenabstände (auch bekannt als \nonfrenchspacing). Für unbekannte Sprachen werden alle Kontexte zurückgesetzt.

7 Erneute Betrachtung der Funktion Letterspacing

\textls [⟨*amount*⟩] {⟨*general text*⟩}

\textls*
\lsstyle Während die tracking-Funktion, die im Abschnitt 5.3 beschrieben wurde, für Sätze von Schriftarten gilt, können sie letterspace (Sperrung) benutzen, um kürzere Stücke des Textes zu erzeugen, unabhängig von der Schriftart in dem sich der Schriftsatz befindet.[13] Für solche Ad-hoc-Zeichenabstände, besitzt microtype zwei Befehle, (unabhängig davon, ob die tracking-Option aktiviert ist), die auf die gleiche Weise verwendet werden können wie die LaTeX-Befehle: \textls – welches auch im Mathemodus funktioniert – erwartet Text im notwendigen Argument, während \lsstyle 'letterspacing' für alle nachfolgenden Schriften bis zum Ende der aktuellen Gruppe aktiviert wird. Die Favoriten-Version von \textls fügt keine zusätzlichen Buchstabenabstände, vor oder nachdem Text ein, was z. B. für Absatztitel nützlich sein kann. Standardmäßig wird jedes Zeichen mit 100/1000 em = 0.1 em von einander getrennt, dieser Betrag kann im optionalen Argument von \textls, durch Verwendung des \SetTracking Befehl oder global über die letterspace Paketoption, geändert werden, mit abnehmender Signifikanz in dieser Reihenfolge.

\lslig {⟨*ligature*⟩}

Seitdem die Befehle \textls und \lsstyle auch die 'no ligatures'-Schlüssel für die entsprechenden Schriftarten auswerten, brauchen Sie sich keine Gedanken mehr über das Schützen oder Trennen von Liganten in den meisten Schriftarten

---

[13] Letterspacing sollte vorsichtig verwendet werden, insbesondere der Zeichenabstand der Kleinbuchstaben wird mit Abscheu von einigen Typpgrafen betrachtet. Falls sie wissen was sie tun, sollten Sie vielleicht kleine Kapitel oder oder alle Kapitel nutzen. Eine andere Eigenschaft kann zur Betonung in Texten von Schriftarten verwendet werden.

zu machen. Jedoch, in bestimmten Situationen, kann es zu einem Konflikt der Liganten, welche mit den gleichen Buchstaben beginnen, kommen, wobei einige von diesen verhindert werden sollten, während das bei anderen nicht geschehen soll. Beim letterspacing von Texten mit Textschriftsatz in Frakturschriftarten, sollen zum Beispiel, die Liganten 'ch', 'ck', 'tz' und 'sz' ('ß') niemals getrennt werden, normalerweise sieht man auch 'St.' ('St.') Liganten im letterspacing-Text. Darüber hinaus realisiert das yfonts Paket das kurze s ('ſ') als den Liganten 's:' andererseits, wenn der 'ct' Ligant und die anderen 'lang s' Liganten in Frakturschriftarten gefunden werden, unterdrückt werden. Es gibt zwei Wege dieses Problem zu lösen: Entweder deaktivieren Sie nicht die „ s und/ oder c "Ligaturen und trennen diejenigen, die getrennt werden müssen durch das Einfügen von '{\kern0pt}' oder babel's Abkürzung oder deaktivieren Sie diese und schützen jene Liganten, welche geschützt werden müssen, durch umschließen mit dem \lslig Befehl. Damit liefern folgende Lösungen das gleiche Ergebnis (nämlich, 'Aussichtslosigkeit').

```
\SetTracking[no ligatures={f}]{encoding = LY, family = yfrak}{100}
\textfrak{\lsstyle Aus:s{\kern0pt}ichts:los{\kern0pt}igkeit}
```

```
\SetTracking[no ligatures={f,s,c}]{encoding = LY, family = yfrak}{100}
\textfrak{\lsstyle Au\lslig{s:}si\lslig{ch}t\lslig{s:}losigkeit}
```

letterspace.sty Diese drei Befehle (mitsamt der letterspace-Option, beschrieben in Abschnitt 3.4) sind auch in dem alternativen letterspace Paket enthalten, welches an sich eine sehr stark abgespeckte Version von microtype ist, das eine Unterstützung für alle anderen Erweiterungen weglässt (zudem wurden die Möglichkeiten des \SetTracking Befehls weggelassen und alle 'f' Ligaturen werden deaktiviert, innere und äußere Abstände und der äußere Buchstabenabstand wurden auf ihre Standardwerte zurückgesetzt, so wie in Abschnitt 5.3 beschrieben). Wenn Sie es bevorzugen auf microtype's Besonderheiten zu verzichten, können Sie stattdesen das letterspace Paket laden. Beide Pakete können nicht zur gleichen Zeit benutzt werden.

Im Gegensatz zu microtype, welches LaTeX verlangt, arbeitet das letterspace Paket auch mit eplain oder auch nur mit miniltx: Beim Gebrauch von eplain laden Sie das Paket mit \usepackage innerhalb \beginpackages ... \endpackages Umgebung, für miniltx (welches die Paketoptionen nicht unterstützt) einfach mit \input letterspace.sty laden.

8 Deaktivieren von Ligaturen

\DisableLigatures [⟨*characters*⟩] {⟨*set of fonts*⟩}

Beim vollständigen Deaktivieren aller Liganten einer Schriftart (welches auch Kerning für diese Schriftart ausschaltet, senkt *lowers* absichtlich die mikro-typografische Qualität des Textes, anstatt diese zu erhöhen, dies ist speziell für Schreibmaschine-Schriftarten nützlich, so dass, z. B. in einer T1 kodierten Schriftart, '\texttt{--}' natürlich als '--', nicht als '-'gedruckt wird. \DisableLigatures kann dazu verwendet werden, um auf gewöhnlichem Weg eine Reihe von Schriftarten zu spezifizieren, für welche Liganten deaktiviert sein sollen, z. B. für die Schreibmaschinenschrift kodiert in T1:

```
\DisableLigatures{encoding = T1, family = tt* }
```

Es ist auch möglich nur ganz bestimmt Liganten zu deaktivieren. Das optionale
Argument, kann eine durch Komma getrennte Liste von Zeichen enthalten, für die
der Ligantenmechanismus unterbunden werden soll:

```
\DisableLigatures[?,!]{encoding = T1} % inhibit ?' and !', but not fi, –, », etc.
```

Dabei ist der Buchstabe, mit dem der Ligant beginnt von Bedeutung. Dieser Befehl
kann nur in der Präambel verwendet werden und auch nur einmal. Der Befehl
fordert pdfTEX 1.30 oder eine aktuellere Version.

9 Hinweise und Warnungen

Verwenden Sie Einstellungen, die zu Ihrer Schriftart passen. Obwohl die Standardeinstellungen akzeptable Ergebnisse für die meisten Schriften liefern sollten, kann
es sein, dass die spezielle Schriftart, welche Sie gerade verwenden, verschiedene
Schriftformen hat, sodass mehr oder weniger Hervorhebung oder Vergrößerungen erforderlich sind. Insbesondere kursive Schriftformen können sich deutlich
in unterschiedlichen Schriftarten unterscheiden, deshalb habe ich mich dagegen
entschieden Standardhervorhebungseinstellungen für diese anzubieten. Die Datei
`test-microtype.tex` kann eine Hilfe, beim Anpassen der Hervorhebungseinstellungen für eine Schriftart sein.

Verwenden Sie keinen zu großen Wert für die Vergrößerung. Font-expansion
(Schriftartenvergrößerung) ist eine Funktion, welche die typographische Qualität des Dokuments verbessern soll, indem sie eine gleichmäßige Graustufe des
Textblocks erzeugt (und potenziell die Verringerung der Zahl der erforderlichen
Silbentrennungen). Wenn die Erweiterung oder Verkleinerung einer Schrift zu groß
oder zu klein wird, so wird der Effekt ins Gegenteil gewandelt. Die Vergrößerung
der Schriftarten um mehr als 2%, d. h. der Wert der Einstellung `Strecken` größer
als 20 ist, erfordert ein geschultes Auge. Wenn Sie das Glück haben und im Besitz von mehreren Kopien einer multiplen Master-Schriftart sind, können Sie die
Vergrößerungsgrenze bis auf 4% erhöhen.

*Verwenden Sie keine Schriftartvergrößerung für Web-Dokumente (mit älteren
pdfTEX Versionen).* Mit pdfTEX Versionen die älter als die 1.40 Version sind, wird
jede vergrößerte Kopie der Schriftart in der PDF-Datei eingebettet, folglich kann
die Dateigröße stark zunehmen (abhängig von Vergrößerungsgrenze und -schritt).
Deshalb, für die Zuvorkommenheit und für Sparsamkeit der Bandbreite, aktivieren
Sie nicht die Schriftartenvergrößerung, beim Erstellen von Dateien, die elektronisch
verteilt werden. Mit pdfTEX 1.40, welches eine andere Technik der Vergrößerung
nutzt, kann die Erhöhung der Dateigröße vernachlässigt werden.

Möglicherweise möchten sie Hervorhebungen im Inhaltsverzeichnis deaktivieren.
In unglücklichen Fällen könnte die aktivierte Hervorhebung, die Längenlinie im
Inhaltsverzeichnis und ähnliche Listen auf eine Weise verändern, sodass ein Excess-
Leader-Punkt eingefügt werden muss. Die Lösung ist, Hervorhebungen vorübergehend im Inhaltsverzeichnis zu deaktivieren:

```
\microtypesetup{protrusion=false}
\tableofcontents
\microtypesetup{protrusion=true}
```

Möglicherweise möchten Sie die Hervorhebungen in der `verbatim` *Umgebung deaktivieren.* Wie Sie bereits wissen, wird microtype standardmäßig die Zeichenhervorhebung in allen Schriftarten aktiviert, die im Schriftartenset 'alltext' enthalten sind. Das schließt auch die Schreibmaschinenschrift mit ein. Obwohl es wirklich Sinn macht, die Schreibmaschine-Schriftart hervorzuheben, wenn Sie im laufenden Text erscheint (wie zum Beispiel in diesem Handbuch), ist dies wahrscheinlich nicht wünschenswert innerhalb der `verbatim` Umgebung. Jedoch hat microtype kein Wissen über den Kontext, in dem eine Schriftart steht, dies wird allein durch die Bestimmung der Attribute entschieden. Deshalb müssen Sie sich um die Deaktivierung der Hervorhebung in der `verbatim` Umgebung selbst kümmern (d. h. wenn Sie Hervorhebungen in der Schreibmaschinenschrift nicht durch auswählen eines anderen Schriftsatzes insgesamt deaktivieren wollen). Obwohl der `\microtypesetup` Befehl natürlich für Fälle wie diesen entwickelt wurde, könnten Sie es anstrengend finden, ihn jedesmal zu wiederholen, wenn Sie ihn in der `verbatim` Umgebung öfters nutzen. Die folgende Zeile, die der Präambel hinzugefügt wird, hätte den selben Effekt:

```
\g@addto@macro\@verbatim{\microtypesetup{activate=false}}
```

Falls sie das fancyvrb oder das listings Paket verwenden, ist dies nicht notwenig, da diese so konzipiert sind, das Sie in der entsprechenden Umgebung, die Hervorhebung in jedem Fall hemmen.

Einstellungen für Griechisch/Thailändisch/Amerikanisch usw. Kodierungen sind bis jetzt noch nicht inbegriffen. Die Standardsätze von Schriftarten, für welche die mikro-typografischen Funktionen aktiviert werden (siehe Tabelle 2), enthalten nur die Kodierungen, für die auch Konfigurationen existieren. Deshalb, wenn Sie eine andere Kodierung verwenden (z. B. LGR, T2B, etc.), wird microtype nicht auf diese Schriftarten anwendbar sein. Sie müssen einen neuen Schriftsatz, einschließlich der Kodierung, definieren und aktivieren, um Sie benutzen zu können(weitere Details, siehe Abschnitt 4). Spätestens für die Hervorhebungen müssten Sie Einstellungen für die Schriftarten, die in Frage kommen, erstellen (siehe Abschnitt 5.1). Selbstverständlich sind Beiträge für diese Art von Kodierung mehr als willkommen.

Nehmen Sie nur dann Kerning-Anpassungen vor, wenn es üblich in der typografischen Tradition der Sprache ist. Im Gegensatz zu Hervorhebung und der Vergrößerung trägt zusätzliches Kerning nicht unbedingt zur Verbesserung der mikro-typographische Qualität des Dokuments bei. Sie sollten das zusätzliche Kerning nur aktivieren, wenn Sie ein Dokument in einer Sprache schreiben, dessen typografische Tradition ein solches Kerning garantiert. Wenn Sie zum Beispiel einen englischen Text schreiben, würden die Leser durch zusätzliche Leerzeichen vor dem Satzzeichen verwirrt werden.

Anpassung der Zwischenwortabstandes befindet sich noch in einem experimentellen Stadium. Die Umsetzung dieser Eigenschaft in pdfTeX ist nicht vollständig und

kann zu den positiven Auswirkungen auf die typographische Qualität, die Sie eventuell erwarten, nicht beitragen. In bestimmten Situationen kann es sogar zu unerwünschten Nebenwirkungen kommen. Deshalb sollte die `Abstand`option nicht blind gewählt werden. Es wird empfohlen, mit den Einstellungen zu experimentieren, um zu verstehen, wie die Eigenschaften funktionieren.

Kompatibilität und Interaktion mit anderen Paketen: Das `microtype` -Paket sollte sehr gut mit anderen LATEX-Paketen zusammen arbeiten (ausgenommen von `pdfcprot`). Jedoch, da das Leben nicht perfekt ist, sind Probleme zu erwarten. Momentan sind folgende Probleme bekannt:

- Wenn Sie 8-Bit-Zeichen in der Konfiguration verwenden möchten, müssen Sie zuerst das Paket `inputenc` laden. Das Verwenden von Unicode wird ebenfalls unterstützt (wenn Sie `inputenc` mit der `utf8` oder `utf8x` Option laden). Bei der Verwendung von mehreren Input-Kodierungen in einem Dokument werden 8-Bit-Zeichen in den Einstellungen nur zuverlässig funktionieren, wenn Sie den `inputenc`-Schlüssel spezifizieren.

- Beim Laden des Pakets mit der `babel`-Option, sollten Sie das `babel`-Paket vor `microtype` laden.

- Es ist derzeit nicht möglich, Charakter-spezifische Einstellungen für chinesische/ japanischen/koreanischen Schriften zu erstellen. Deshalb, ist die einzige mikro-typografische Erweiterung, die mit dem `CJK` Paket arbeiten kann, die Schriftart-vergrößerung.

Mögliche Fehlermeldungen und wie man diese beseitigt:

- `! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.`
  Diese Fehlermeldung tritt auf, wenn Sie versuchen die Schriftartvergrößerung zu verwenden, während Sie eine DVI-Ausgabe erstellen. Denken Sie daran, dass die *automatische* Schriftartenerweiterung nur als pdfTEX im PDF-Modus funktioniert. Vergrößerung ist auch im DVI-Modus möglich. Dies erfordert, dass alle Instanzen der erweiterten Schriftarten auf Ihrem TEX-System existieren.

- `! pdfTeX error (font expansion): auto expansion is only possible with scalable fonts.`
  Die automatische Schriftartvergrößerung ist in pdfTEX 1.40 verbessert worden, dass sie nicht nur mit Typ 1 Schriften sondern auch mit TrueType, OpenType und sogar auch mit nicht-eingebetteten Schriftarten arbeitet. Die obige Fehlermeldung bedeutet entweder, dass Sie versuchen, die Vergrößerung auf eine Bitmap (`pk`) Schriftart anzuwenden, die noch nicht möglich ist, oder dass die Schrift nicht gefunden werden kann, da z. B. deren Einträge fehlen.

- `Warning: pdflatex: font ptmr8r cannot be expanded (not an included Type1 font)`
  und der PDF-Viewer beklagt sich über eine fehlende Schriftart, z. B der Adobe Reader auf diese Weise:
  `Could not find a font in the Resources dictionary - using Helvetica instead.`
  Mit pdfTEX Versionen, die älter sind als 1.40, kann Schriftartvergrößerung nur dann angewandt werden, wenn die Schriftart in der PDF-Datei eingebettet wird. Falls Sie die obengenannte Fehlermeldung erhalten, kann Ihr TEX-System noch nicht mit PostScript-Schriftarten (z. B. Times, Helvetica, Courier) arbeiten, da diese in die Basisdatei eingebettet werden müssen. In den meisten TEX-Distributionen, kann

diese in der Datei `updmap.cfg` durch Einstellungen der `pdftexDownloadBase14` in `true` geändert werden.

- `Warning: pdflatex (file ecrm1000+20): Font ecrm1000+20 at 1200 not found`

  Außerdem erfordern pdfTeX-Versionen, die älter als 1.40 sind, Typ 1-Schriften für die automatische Schriftarten-Vergrößerung. Wenn Sie eine Nachricht wie die obengenannte erhalten, haben Sie wahrscheinlich versucht, die Schriftartvergrößerung auf eine bitmap oder TrueType Schriftart anzuwenden. Bei älteren pdfTeX-Versionen ist dies nur möglich, wenn Sie manuell erweiterte Instanzen der Schriften erstellen.

- `! Font T1/cmr/m/n/10=ecrm1000 at 10.0pt not loaded: Not enough room left.`

  Speicherparameter 'font_mem_size' zu klein.

- `! TeX capacity exceeded, sorry [maximum internal font number (font_max)=2000].`

  Speicherparameter 'font_max' zu klein.

- `! TeX capacity exceeded, sorry [PDF memory size (pdf_mem_size)=65536].`

  Speicherparameter 'pdf_mem_size' zu klein (bei pdfTeX Versionen, die älter sind als 1.30).

  Bei Anwendung von mikro-typografischen Erweiterung eines großes Dokumentes mit einer Menge von Schriftarten, kann die Speicherkapazität von pdfTeX knapp werden. Dies kann durch Setzen der entsprechenden Parameter, auf einen größeren Wert erhöht werden. Für web2c-basierte Systeme, z. B. TeX Live, sollten Sie die Einstellungen in der Datei `texmf.cnf`, und für MiKTeX in der Datei `miktex.ini` (für 2,4 Jahre ältere Versionen) bzw. `pdflatex.ini` (2,5 Jahre neuere Versionen) ändern.

- `pdfTeX warning (font expansion): font should be expanded before its first use`

  Diese Warnung wird bei den pdfTeX-Versionen vorkommen, die älter sind als 1.40.4, wenn Tracking *und* Vergrößerung auf eine Schriftart angewendet werden. Sie ist harmlos und kann ignoriert werden.

10 Beiträge

Ich wäre froh, Konfigurationsdateien für mehr Schriftarten beifügen zu können. Das Vorbereiten solcher Konfigurationen ist recht zeitaufwändig und erfordert viel Geduld. Um diesen Prozess zu erleichtern, schließt dieses Paket auch eine Testdatei ein, die verwendet werden kann, um zumindest die Einstellungen zur Hervorhebung überprüfen zu können (`test-microtype.tex`).

Wenn Sie eine Konfigurationsdatei für eine andere Schriftart erstellt oder Anregungen für Verbesserungen in den Standardkonfigurations-Dateien haben, würde ich diese dankbar entgegennehmen: w.m.l@gmx.net.

11 Danksagungen

Dieses Paket wäre sinnlos, wenn *Hàn Thế Thành* dieses pdfTeX-Programm überhaupt nicht erstellt hätte, welches die mikro-typographische Erweiterung eingeführt und so für die TeX-Welt verfügbar machte. Darüber hinaus danke ich ihm für die

Hilfe, dieses Paket zu verbessern und nicht zuletzt für die Förderung in Thành 2004 und Thành 2008 und anderswo. Also bedanke ich mich bei ihm und dem Rest der pdfTEX Mannschaft für die Widerlegung der Behauptung, dass TEX tot sei, und für das Fixen der Programmfehler, die ich gefunden habe.

Harald Harders hat dazu beigetragen, die Vorsprungseinstellungen für Adobe Minion möglich zu machen. Ich möchte ihm auch für eine Reihe an Fehlermeldungen und Vorschlägen danken, die er gemacht hat. *Andreas Bühmann* hat die Möglichkeit vorgeschlagen, Bereiche von Schriftgrößen anzugeben und geistreich bei der Implementierung dieser zu helfen. Er hatte auch einige guten Ideen für das Management von komplexen Konfigurationen. *Ulrich Dirr* hat zahlreiche Vorschläge gemacht, besonders bezüglich der neuen Erweiterungen der Zwischenwortabstandsanpassung und zum zusätzlichen Kerning. Mein Dank gilt auch *Maciej Eder* für die beigetragenen Einstellungen der QX-Kodierung, sowie *Karl Karlsson* zur Bereitstellung von Einstellungen für die kyrillische T2A-Kodierung. Ich bedanke mich bei *Élie Roux*, dass sie das `lua`-Modul erschaffen hat.

Ich danke *Philipp Lehman*, dass er seinem `csquotes`-Paket die Möglichkeit hinzugefügt hat, die ursprüngliche Bedeutung aller aktivierten Zeichen wiederherzustellen, sodass diese für die Zeichen in den Konfigurationsdateien verwendet werden können. Freundlicherweise hat *Peter Wilson* ein Häkchen in seinen `ledmac`/`ledpar`-Paketen zur Verfügung gestellt, sodass kritische Editionen schließlich auch vom Charakter-Vorsprung profitieren.

Zusätzlich haben die folgenden Personen Bugs gemeldet, Vorschläge gemacht oder anders geholfen (in chronologischer Reihenfolge): *Tom Kink, Herb Schulz, Michael Hoppe, Gary L. Gray, Georg Verweyen, Christoph Bier, Peter Muthesius, Bernard Gaulle †, Adam Kucharczyk, Mark Rossi, Stephan Hennig, Michael Zedler, Herbert Voß, Ralf Stubner, Holger Uhr, Peter Dyballa, Morten Høgholm, Steven Bath, Daniel Flipo, Michalis Miatidis, Sven Naumann, Ross Hetherington, Geoff Vallis, Steven E. Harris, Karl Berry, Peter Meier, Nathan Rosenblum, Wolfram Schaalo, Vasile Gaburici, Sveinung Heggen, Colin Rourke, Maverick Woo, Silas S. Brown, Christian Stark* and *Marcin Borkowski*.

12 Literatur

Hàn Thế Thành, 'Micro-typographic extensions to the TEX typesetting system', Diss. Masaryk University Brno 2000, in: *TUGBoat*, vol. 21 (2000), no. 4, pp. 317–434. (Online at http://www.tug.org/TUGboat/Articles/tb21-4/tb69thanh.pdf)

Hàn Thế Thành, 'Micro-typographic extensions of pdfTEX in practice', in: *TUGBoat*, vol. 25 (2004), no. 1: 'Proceedings of the Practical TEX 2004 Conference', pp. 35–38. (Online at http://www.tug.org/TUGboat/Articles/tb25-1/thanh.pdf)

Hàn Thế Thành, 'Font-specific issues in pdfTEX', in: *TUGBoat*, vol. 29 (2008), no. 1: 'EuroBachoTEX 2007 Proceedings', pp. 36–41. (Online at http://www.tug.org/TUGboat/Articles/tb29-1/tb91thanh-fonts.pdf)

Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, Martin Schröder, *The pdfTEX user manual*, 25 January 2007. (Available from CTAN at /systems/pdftex/; latest version at http://sarovar.org/projects/pdftex/)

Karl Berry, *Fontname. Filenames for TEX fonts*, July 2009. (Available from CTAN at `/info/fontname/fontname.pdf`)

LATEX3 Project Team, *LATEX 2ε font selection*, 27 November 2005. (Available from CTAN at `/macros/latex/doc/fntguide.pdf`)

Carsten Schurig, Tobias Schlemmer, *The `pdfcprot.sty` package*, 10 June 2005. (Available from CTAN at `/macros/latex/contrib/pdfcprot/`)

Melchior Franz, *The soul package*, 17 November 2003. (Available from CTAN at `/macros/latex/contrib/soul/`). See also Heiko Oberdiek's extension of this package, soulutf8, which adds Unicode support. (Available from CTAN at `/macros/latex/contrib/oberdiek/`)

13 Kurzer Rückblick

The comprehensive list of changes can be obtained by running '`makeindex -s gglo.ist -o microtype.gls microtype.glo`'.

Die folgende Zusammenstellung ist eine Liste aller Änderungen für das relevante Nutzerland. Fehler im Softwareprogramm und Kompatibilitätsfeststellungen wurden vernachlässigt. Zahlen in eckigen Klammern verweisen auf die relevanten Abschnitte in dieser Dokumentation.

2.4 (2010/01/10)
- `lua` functions moved to a dedicated file
- Protrusion settings for T2A encoded Minion

2.3e (2009/11/09)
- Support for the Cyrillic T2A encoding (protrusion, expansion, spacing)

2.3d (2009/03/27)
- New default for expansion option '`step`': 1, if pdfTEX $\geq$ 1.40 [3.3]

2.3c (2008/11/11)
- Support for luaTEX enabled by default

2.3 (2007/12/23)
- New key '`outer kerning`' for `\SetTracking` to customise outer kerning [5.3]
- Adjust protrusion settings for tracking even if protrusion is not enabled
- New option '`verbose=silent`' to turn all warnings into mere messages [3.5]
- The letterspace package also works with eplain or miniltx [7]

2.2 (2007/07/14)
- Improvements to tracking/letterspacing: retain kerning (pdfTEX $\geq$ 1.40.4); automatically adjust protrusion settings
- New key '`no ligatures`' for `\SetTracking` to disable selected or all ligatures (pdfTEX $\geq$ 1.40.4) [5.3]
- New keys '`spacing`' and '`outer spacing`' for `\SetTracking` to customise interword spacing [5.3]
- Possibility to expand a font with different parameters (pdfTEX $\geq$ 1.40.4) [5.2]

- New optional argument for \DisableLigatures to disable selected ligatures only [8]
- New command \DeclareMicrotypeVariants to specify variant suffixes [5.7]
- New command \textmicrotypecontext as a wrapper for \microtypecontext [6]
- Protrusion settings for Bitstream Letter Gothic

2.1 (2007/01/21)

- New command \lslig to protect ligatures in letterspaced text [7]

2.0 (2007/01/14)

- Support for the new extensions of pdfTeX $\geq$ 1.40: tracking/letterspacing, adjustment of interword spacing (glue), and additional kerning (new commands \SetTracking, \SetExtraSpacing, \SetExtraKerning; new options 'tracking', 'spacing', 'kerning') [5.3, 5.5, 5.4]
- New commands \textls and \lsstyle for letterspacing, new option 'letterspace' [3.4, 7]
- New option 'babel' for automatic micro-typographic adjustment to the selected language [3.5, 6]
- New font sets: 'smallcaps', 'footnotesize', 'scriptsize' [4, table 2]
- New package 'letterspace' providing the commands for robust and hyphenatable letterspacing [7]

1.9e (2006/07/28)

- New key 'inputenc' to specify the lists' input encodings [5]
- Protrusion settings for Euler math fonts

1.9d (2006/05/05)

- Support for the Central European QX encoding (protrusion, inheritance)
- Protrusion settings for various Euro symbol fonts (Adobe, ITC, marvosym)
- Support for Unicode input in the configuration (inputenc/utf8)

1.9c (2006/02/02)

- Protrusion settings for URW Garamond

1.9a (2005/12/05)

- Defer setup until the end of the preamble
- Inside the preamble, \microtypesetup accepts all package options [3.6]
- Protrusion settings for T5 encoded Charter

1.9 (2005/10/28)

- New command \DisableLigatures to disable ligatures of fonts (pdfTeX $\geq$ 1.30) [8]
- New command \microtypecontext to change the configuration context; new key 'context' for the configuration commands [6]
- New key 'font' to add single fonts to the font sets [4]
- New key 'preset' to set all characters to the specified value before loading the lists
- Value 'relative' renamed to 'character' for 'unit' keys
- Support for the Polish OT4 encoding (protrusion, expansion, inheritance)
- Support for the Vietnamese T5 encoding (protrusion, expansion, inheritance)

1.8 (2005/06/23)

- New command \DeclareMicrotypeSetDefault to declare the default font sets [4]
- New option 'config' to load a different configuration file [3.5]
- New option 'unit' to measure protrusion factors relative to a dimension instead of the character width [5.1]
- Renamed commands from \..MicroType.. to \..Microtype..
- Protrusion settings for AMS math fonts
- Protrusion settings for Times in LY1 encoding completed
- The 'allmath' font set also includes U encoding
- When using the ledmac package, character protrusion will work for the first time ever (pdfTEX ≥ 1.30)

1.7 (2005/03/23)

- Possibility to specify ranges of font sizes in the set declarations and protrusion and expansion settings [4, 5]
- New command \LoadMicrotypeFile to load a font configuration file manually [5.7]
- Hook \Microtype@Hook for font package authors [14.4.3]
- New option 'verbose=errors' to turn all warnings into errors
- Warning when running in draft mode

1.6 (2005/01/24)

- New option 'factor' to influence protrusion resp. expansion of all characters of a font or font set [3.2, 5]
- When pdfTEX is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled [3.1]
- Use e-TEX extensions, if available

1.5 (2004/12/15)

- When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled [3.1]
- New option 'selected' to enable selected expansion, default: false [3.3, 5.2]
- New default for expansion option 'step': 4 (min(stretch,shrink)/5) [3.3]
- Protrusion settings for Bitstream Charter

1.4 (2004/11/12)

- Set up fonts independently from LATEX font loading
- New option: 'final' [3.5]

1.2 (2004/10/03)

- New font sets: 'allmath' and 'basicmath' [4, table 2]
- Protrusion settings for Computer Modern Roman math symbols
- Protrusion settings for TS1 encoding completed for Computer Modern Roman and Adobe Garamond

1.1 (2004/09/21)

- Protrusion settings for Adobe Minion
- New command: \DeclareCharacterInheritance [5.6]
- Characters may also be specified as octal or hexadecimal numbers [5]

1.0 (2004/09/11)

- First CTAN release

14 Implementation

The docstrip modules in this file are:

driver: The documentation driver, only visible in the dtx file.

package: The code for the microtype package (microtype.sty).

letterspace: The code for the letterspace package (letterspace.sty).

lua: Code for luaTeX (microtype only).

plain: Code for eplain, miniltx (letterspace only).

debug: Code for additional output in the log file.
Used for – surprise! – debugging purposes.

config: Surrounds all configuration modules.

cfg-t: Surrounds (Latin) text configurations.

m-t: The main configuration file (microtype.cfg).
bch: Settings for Bitstream Charter (mt-bch.cfg).
blg: Settings for Bitstream Letter Gothic (mt-blg.cfg).
cmr: Settings for Computer Modern Roman (mt-cmr.cfg).
pad: Settings for Adobe Garamond (mt-pad.cfg).
ppl: Settings for Palatino (mt-ppl.cfg).
ptm: Settings for Times (mt-ptm.cfg).
pmn: Settings for Adobe Minion (mt-pmn.cfg).
Contributed by *Harald Harders*.
ugm: Settings for URW Garamond (mt-ugm.cfg).

cfg-u: Surrounds non-text configurations (U encoding).

msa: Settings for AMS 'a' symbol font (mt-msa.cfg).
msb: Settings for AMS 'b' symbol font (mt-msb.cfg).
euf: Settings for Euler Fraktur font (mt-euf.cfg).
eur: Settings for Euler Roman font (mt-eur.cfg).
eus: Settings for Euler Script font (mt-eus.cfg).

cfg-e: Surrounds Euro symbol configurations.

zpeu: Settings for Adobe Euro symbol fonts (mt-zpeu.cfg).
euroitc: Settings for ITC Euro symbol fonts (mt-euroitc.cfg).
mvs: Settings for marvosym Euro symbol (mt-mvs.cfg).

test: A helper file that may be used to create and test protrusion settings
(test-microtype.tex).

And now for something completely different.

1 ⟨∗package | letterspace⟩

14.1 Preliminaries

\MT@MT This is us.

```
2 \def\MT@MT
3 ⟨package⟩ {microtype}
4 ⟨letterspace⟩ {letterspace}
```

\MT@fix@catcode We have to make sure that the category codes of some characters are correct (the german package, for instance, makes " active). Probably overly cautious. Ceterum censeo: it should be forbidden for packages to change catcodes within the preamble.

\MT@restore@catcodes Polite as we are, we'll restore them afterwards.

```
5 \let\MT@restore@catcodes\@empty

6 \def\MT@fix@catcode#1#2{%
7   \edef\MT@restore@catcodes{%
8     \MT@restore@catcodes
9     \catcode#1 \the\catcode#1\relax
10   }%
11   \catcode#1 #2\relax
12 }
13 ⟨package⟩\MT@fix@catcode{17}{14}% ^^Q (comment)
14 \MT@fix@catcode{24} {9}% ^^X (ignore)
15 ⟨package⟩\MT@fix@catcode{33}{12}% !
16 ⟨package⟩\MT@fix@catcode{34}{12}% "
17 \MT@fix@catcode{36} {3}% $ (math shift)
18 \MT@fix@catcode{39}{12}% '
19 \MT@fix@catcode{42}{12}% *
20 \MT@fix@catcode{43}{12}% +
21 \MT@fix@catcode{44}{12}% ,
22 \MT@fix@catcode{45}{12}% -
23 \MT@fix@catcode{58}{12}% :
24 \MT@fix@catcode{60}{12}% <
25 \MT@fix@catcode{61}{12}% =
26 \MT@fix@catcode{62}{12}% >
27 ⟨package⟩\MT@fix@catcode{63}{12}% ?
28 \MT@fix@catcode{94} {7}% ^ (superscript)
29 \MT@fix@catcode{96}{12}% `
30 ⟨package⟩\MT@fix@catcode{124}{12}% |
```

These are all commands for the outside world. We define them here as blank commands, so that they won't generate an error if we are not running pdfTeX.

```
31 ⟨*package⟩
32 \newcommand*\DeclareMicrotypeSet[3][]{}
33 \newcommand*\UseMicrotypeSet[2][]{}
34 \newcommand*\DeclareMicrotypeSetDefault[2][]{}
35 \newcommand*\SetProtrusion[3][]{}
36 \newcommand*\SetExpansion[3][]{}
37 \newcommand*\SetTracking[3][]{}
38 \newcommand*\SetExtraKerning[3][]{}
39 \newcommand*\SetExtraSpacing[3][]{}
40 \newcommand*\DisableLigatures[2][]{}
41 \newcommand*\DeclareCharacterInheritance[3][]{}
42 \newcommand*\DeclareMicrotypeVariants[1]{}
43 \newcommand*\DeclareMicrotypeAlias[2]{}
44 \newcommand*\LoadMicrotypeFile[1]{}
45 \newcommand*\DeclareMicrotypeBabelHook[2]{}
46 \newcommand*\microtypesetup[1]{}
47 \newcommand*\microtypecontext[1]{}
48 \newcommand*\textmicrotypecontext[2]{#2}
```

```
49 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
50 ⟨/package⟩
51 \newcommand*\lsstyle{}
52 \newcommand\textls[2][]{}
53 \def\textls#1#{}
54 \newcommand*\lslig[1]{#1}
55 ⟨*package⟩
56 }
```

These commands also have a starred version.

```
57 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
58 \def\DeclareMicrotypeVariants#1#{\@gobble}
```

Set declarations are only allowed in the preamble (resp. the main configuration file). The configuration commands, on the other hand, must be allowed in the document, too, since they may be called inside font configuration files, which, in principle, may be loaded at any time.

```
59 \@onlypreamble\DeclareMicrotypeSet
60 \@onlypreamble\UseMicrotypeSet
61 \@onlypreamble\DeclareMicrotypeSetDefault
62 \@onlypreamble\DisableLigatures
63 \@onlypreamble\DeclareMicrotypeVariants
64 \@onlypreamble\DeclareMicrotypeBabelHook
```

\MT@old@cmd   The old command names had one more hunch.

```
65 \def\MT@old@cmd#1#2{%
66   \newcommand*#1{\MT@warning{%
67     \string#1 is deprecated. Please use\MessageBreak
68     \string#2 instead}%
69     \let #1#2#2}}
70 \MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
71 \MT@old@cmd\DeclareMicroTypeSet   \DeclareMicrotypeSet
72 \MT@old@cmd\UseMicroTypeSet       \UseMicrotypeSet
73 \MT@old@cmd\LoadMicroTypeFile     \LoadMicrotypeFile
74 ⟨/package⟩
```

\MT@warning   Communicate.
\MT@warning@nl

```
75 \def\MT@warning{\PackageWarning\MT@MT}
76 \def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
77 ⟨*package⟩
78 \def\MT@info{\PackageInfo\MT@MT}
79 \def\MT@info@nl#1{\MT@info{#1\@gobble}}
80 \let\MT@vinfo\@gobble
81 \def\MT@error{\PackageError\MT@MT}
82 \def\MT@warn@err#1{\MT@error{#1}{%
83   This error message appears because you loaded the ‘\MT@MT’\MessageBreak
84   package with the option ‘verbose=errors’. Consult the documentation\MessageBreak
85   in \MT@MT.pdf to find out what went wrong.}}
```

\MT@info
\MT@info@nl
\MT@vinfo
\MT@error
\MT@warn@err

14.1.1 Debugging

\tracingmicrotype   Cases for \tracingmicrotype:
\MT@dinfo
\MT@dinfo@nl

0: almost none

1: + sets & lists

2: + heirs

3: + slots

4: + factors

```
86 ⟨∗debug⟩
87 \MT@warning@nl{This is the debug version}
88 \newcount\tracingmicrotype
89 \tracingmicrotype=2
90 \def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
91 \def\MT@info@nl#1{\PackageInfo\MT@MT{#1\@gobble}\MT@addto@annot{#1}}
92 \let\MT@vinfo\MT@info@nl
93 \def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
94 \def\MT@warning@nl#1{\PackageWarning\MT@MT{#1\@gobble}\MT@addto@annot{Warning: #1}}
95 \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
96 \def\MT@dinfo@nl#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info@nl{#2}\fi}
```

\tracingmicrotypeinpdf   Another debug method: font switches can be marked in the PDF file with a small caret, an accompanying popup text box displaying all debug messages.

Cases for \tracingmicrotypeinpdf:

1: show new fonts

2: + show known fonts

```
97 \newcount\tracingmicrotypeinpdf
```

[If `microtype.sty` had been generated with the '`debug`' option, this method would be demonstrated here.]

\MT@pdf@annot
\MT@addto@annot
\ifMT@inannot

During font setup, we save the text for the popup in \MT@pdf@annot. (This requires pdfTEX ≥ 1.30.) The **pdftexcmds** package provides pdfTEX's utility commands in luaTEX, too.

```
 98 \RequirePackage{pdftexcmds}
 99 \newif\ifMT@inannot \MT@inannottrue
100 \let\MT@pdf@annot\@empty
101 \def\MT@addto@annot#1{\ifnum\tracingmicrotypeinpdf>\z@ \ifMT@inannot
102   {\def\MessageBreak{^^J\@spaces}%
103    \MT@xadd\MT@pdf@annot{\pdf@escapestring{#1^^J}}}\fi\fi}
```

\iftracingmicrotypeinpdfall   With \tracingmicrotypeinpdfallfalse, the PDF output is (hopefully) identical, but some font switches will not be displayed; otherwise the output is affected, but *all* font switches are visible. In the latter case, we also insert a small kern so that multiple font switches are discernable.

```
104 \newif\iftracingmicrotypeinpdfall
```

\MT@show@pdfannot   A red caret is shown for fonts which are actually set up by Microtype, a green one marks fonts that we have already seen. The /Caret annotation requires a viewer for PDF version 1.5 (you could use /Text if you're using an older PDF viewer).

```
105 \def\MT@show@pdfannot#1{%
106   \ifnum\tracingmicrotypeinpdf<#1 \else
107     \iftracingmicrotypeinpdfall\leavevmode\fi
108     \pdfannot height 4pt width 4pt depth 2pt {%
109       /Subtype/Caret
110       /T(\expandafter\string\font@name)
111       \ifcase#1\or
112       /Subj(New font)/C[1 0 0]
113       \else
114       /Subj(Known font)/C[0 1 0]
115       \fi
116       /Contents(\MT@pdf@annot)
117     }%
118     \iftracingmicrotypeinpdfall\kern1pt \fi
```

```
119      \global\MT@inannotfalse
120    \fi
121 }
122 ⟨/debug⟩
123 ⟨/package⟩
```

14.1.2   Requirements

\MT@plain   The letterspace package works with:

0:   miniltx

1:   eplain

2:   LATEX

For plain usage, we have to copy some commands from `latex.ltx`.

```
124 ⟨*plain⟩
125 \def\MT@plain{2}
126 \ifx\documentclass\@undefined
127    \def\MT@plain{1}
128    \def\hmode@bgroup{\leavevmode\bgroup}
129    \def\nfss@text#1{{\mbox{#1}}}
130    \let\@typeset@protect\relax
131    \ifx\eplain\@undefined
132      \def\MT@plain{0}
133      \def\PackageWarning#1#2{%
134        \begingroup
135          \newlinechar=10 %
136          \def\MessageBreak{^^J(#1)\@spaces\@spaces\@spaces\@spaces}%
137          \immediate\write16{^^JPackage #1 Warning: #2\on@line.^^J}%
138        \endgroup
139      }
140      \def\on@line{ on input line \the\inputlineno}
141      \def\@spaces{\space\space\space\space}
142    \fi
143 \fi
```

\MT@requires@latex   Better use groups than plain ifs.

```
144 \def\MT@requires@latex#1{%
145    \ifnum\MT@plain<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
146 }
147 ⟨/plain⟩
```

\MT@pdftex@no   pdfTEX's features for which we provide an interface here haven't always been available, and some specifics have changed over time. Therefore, we have to test which pdfTEX we're using, if any. \MT@pdftex@no will be used throughout the package to respectively do the right thing.

Currently, we have to distinguish seven cases for pdfTEX:

0:   not running pdfTEX

1:   pdfTEX (< 0.14f)

2:   + micro-typographic extensions (0.14f,g)

3:   + protrusion relative to 1 em (≥ 0.14h)

4:   + automatic font expansion; protrusion no longer has to be set up first; scale factor fixed to 1000; default \efcode $= 1000$ (≥ 1.20)

5: $+$ \(left,right)marginkern; \pdfnoligatures; \pdfstrcmp; \pdfescapestring (≥ 1.30)

6: $+$ adjustment of interword spacing; extra kerning; \letterspacefont; \pdfmatch[14]; \pdftracingfonts; always e-TeX (≥ 1.40)

7: $+$ \letterspacefont doesn't disable ligatures and kerns; \pdfcopyfont ($\geq 1.40.4$)

```
148 \def\MT@pdftex@no{0}
```

A hack circumventing the TeX Live 2004 hack which undefines the pdfTeX primitives in the format in order to hide the fact that pdfTeX is being run from the user. This has been *fixed* in TeX Live 2005.

```
149 \ifx\normalpdftexversion\@undefined \else
150   \let\pdftexversion \normalpdftexversion
151   \let\pdftexrevision\normalpdftexrevision
152   \let\pdfoutput      \normalpdfoutput
153 \fi
```

Old packages might have let \pdftexversion to \relax.

```
154 \ifx\pdftexversion\@undefined \else
155   \ifx\pdftexversion\relax \else
156 ⟨debug⟩\MT@dinfo@nl{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
157     \def\MT@pdftex@no{7}
158 ⟨*package⟩
159     \ifnum\pdftexversion = 140
160       \ifnum\pdftexrevision < 4
161         \def\MT@pdftex@no{6}
162       \fi
163     \else
164 ⟨/package⟩
165       \ifnum\pdftexversion < 140
166         \def\MT@pdftex@no{5}
167 ⟨*package⟩
168         \ifnum\pdftexversion < 130
169           \def\MT@pdftex@no{4}
170           \ifnum\pdftexversion < 120
171             \def\MT@pdftex@no{3}
172           \ifnum\pdftexversion = 14
173             \ifnum \expandafter`\pdftexrevision < `h
174               \def\MT@pdftex@no{2}
175               \ifnum \expandafter`\pdftexrevision < `f
176                 \def\MT@pdftex@no{1}
177               \fi
178             \fi
179           \else
180             \ifnum\pdftexversion < 14
181               \def\MT@pdftex@no{1}
182             \fi
183           \fi
184         \fi
185       \fi
186     \fi
187 ⟨/package⟩
188     \fi
189   \fi
```

---

14   This command was actually introduced in 1.30, but failed on strings longer than 1023 bytes.

```
190 \fi
191 ⟨debug⟩\MT@dinfo@nl{0}{pdftex no.: \MT@pdftex@no}
```

\MT@clear@options   If we are not using pdfTeX or in case it is too old, we disable everything and exit.

```
192 \def\MT@clear@options{%
193 ⟨plain⟩   \MT@requires@latex1{%
194   \AtEndOfPackage{\let\@unprocessedoptions\relax}%
195   \let\CurrentOption\@empty
196 ⟨plain⟩   }\relax
197 }
198 \ifnum\MT@pdftex@no <
199 ⟨package⟩     2
200 ⟨letterspace⟩      6
201   \MT@warning@nl{You
202     \ifcase\MT@pdftex@no
203       don't seem to be using pdftex.\MessageBreak
204       '\MT@MT' only works with pdftex.\MessageBreak
205       Try running 'pdflatex' instead of
206         '\ifx\XeTeXversion\@undefined\else xe\fi latex'%
207     \else
208       are using a pdftex version older than
209 ⟨package⟩       0.14f%
210 ⟨letterspace⟩       1.40%
211       .\MessageBreak
212       '\MT@MT' does not work with this version.\MessageBreak
213       Please install a newer version of pdftex%
214     \fi
215   }
216   \MT@clear@options\MT@restore@catcodes
217 \endinput\fi
```

Since luaTeX is included in TeX Live 2008, we now support it by default, even though it's still experimental. Letterspacing doesn't work at all yet, since luaTeX doesn't know the \letterspacefont command.

```
218 ⟨*!lua | letterspace⟩
219 \ifx\directlua\@undefined \else
220   \ifx\directlua\relax \else
221 ⟨!letterspace⟩     \MT@error
222 ⟨letterspace⟩      \MT@warning@nl
223     {'\MT@MT'
224 ⟨!letterspace⟩      only works with luatex if you generate%
225 ⟨letterspace⟩      doesn't currently work with luatex.%
226       \MessageBreak
227 ⟨!letterspace⟩      the package with the 'lua' option%
228 ⟨letterspace⟩      Bye bye%
229     }
230 ⟨!letterspace⟩     {}
231 ⟨letterspace⟩      \MT@clear@options\MT@restore@catcodes
232 ⟨letterspace⟩      \expandafter\expandafter\expandafter\endinput
233   \fi
234 \fi
235 ⟨/!lua | letterspace⟩
```

Still there? Then we can begin: We need the keyval package, including the 'new' \KV@@sp@def implementation.

```
236 \RequirePackage{keyval}[1997/11/10]
237 ⟨*package⟩
```

\MT@toks   We need a token register.

238 `\newtoks\MT@toks`

`\ifMT@if@`   A scratch if.

239 `\newif\ifMT@if@`

14.1.3 Declarations

`\ifMT@protrusion`   These are the global switches . . .

`\ifMT@expansion` 240 `\newif\ifMT@protrusion`
`\ifMT@auto` 241 `\newif\ifMT@expansion`
`\ifMT@selected` 242 `\newif\ifMT@auto`
  243 `\newif\ifMT@selected`
`\ifMT@noligatures` 244 `\newif\ifMT@noligatures`
`\ifMT@draft` 245 `\newif\ifMT@draft`
`\ifMT@spacing` 246 `\newif\ifMT@spacing`
`\ifMT@kerning` 247 `\newif\ifMT@kerning`
  248 `\newif\ifMT@tracking`
`\ifMT@tracking` 249 `\newif\ifMT@babel`
`\ifMT@babel`
`\MT@pr@level`   . . . and numbers.

`\MT@ex@level` 250 `\let\MT@pr@level\tw@`
`\MT@pr@factor` 251 `\let\MT@ex@level\tw@`
  252 `\let\MT@pr@factor\@m`
`\MT@ex@factor` 253 `\let\MT@ex@factor\@m`
`\MT@sp@factor` 254 `\let\MT@sp@factor\@m`
`\MT@kn@factor` 255 `\let\MT@kn@factor\@m`

`\MT@pr@unit`   Default unit for protrusion settings is character width, for spacing space, for
`\MT@sp@unit`   kerning (and tracking) 1 em.

`\MT@kn@unit` 256 `\let\MT@pr@unit\@empty`
  257 `\let\MT@sp@unit\m@ne`
  258 `\def\MT@kn@unit{1em}`

`\MT@stretch`   Expansion settings.

`\MT@shrink` 259 `\let\MT@stretch\m@ne`
`\MT@step` 260 `\let\MT@shrink \m@ne`
  261 `\let\MT@step   \m@ne`

`\MT@pr@min`   Minimum and maximum values allowed by pdfTEX.

`\MT@pr@max` 262 `\def\MT@pr@min{-\@m}`
`\MT@ex@min` 263 `\let\MT@pr@max\@m`
  264 `\let\MT@ex@min\z@`
`\MT@ex@max` 265 `\let\MT@ex@max\@m`
`\MT@sp@min` 266 `\def\MT@sp@min{-\@m}`
`\MT@sp@max` 267 `\let\MT@sp@max\@m`
`\MT@kn@min` 268 `\def\MT@kn@min{-\@m}`
`\MT@kn@max` 269 `\let\MT@kn@max\@m`
  270 ⟨/package⟩
`\MT@tr@min` 271 `\def\MT@tr@min{-\@m}`
`\MT@tr@max` 272 `\let\MT@tr@max\@m`
  273 ⟨∗package⟩

`\MT@factor@default` Default factor.

274 `\def\MT@factor@default{1000 }`

`\MT@stretch@default` Default values for expansion.

`\MT@shrink@default` 275 `\def\MT@stretch@default{20 }`
`\MT@step@default` 276 `\def\MT@shrink@default{20 }`
  277 `\def\MT@step@default{4 }`

\MT@letterspace   Default value for letterspacing (in thousandths of 1 em).

\MT@letterspace@default   278 ⟨/package⟩
                      279 `\let\MT@letterspace\m@ne`
                      280 `\def\MT@letterspace@default{100}`
                      281 ⟨∗package⟩

\ifMT@document   Our private test whether we're still in the preamble.

                      282 `\newif\ifMT@document`

14.1.4 Auxiliary macros

\MT@maybe@etex   For definitions that depend on e-TeX features.

                      283 `\ifcase 0%`
                      284   `\ifx\eTeXversion\@undefined 1\else`
                      285   `\ifx\eTeXversion\relax    1\else`
                      286   `\ifcase\eTeXversion     1\fi`
                      287   `\fi`
                      288   `\fi`
                      289 `\else`
                      290   `\catcode`\^^Q=9 \catcode`\^^X=14`
                      291 `\fi`
                      292 ⟨debug⟩`\MT@dinfo@nl{0}{this is`
                      293 ⟨debug⟩`^^Q not`
                      294 ⟨debug⟩`  etex}`

\MT@requires@pdftex   For definitions that depend on a particular pdfTeX version.

                      295 `\def\MT@requires@pdftex#1{%`
                      296   `\ifnum\MT@pdftex@no<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi`
                      297 `}`
                      298 ⟨debug⟩`\MT@requires@pdftex6{\pdftracingfonts=1 }\relax`

\MT@requires@luatex   For definitions that depend on luaTeX.

                      299 ⟨∗lua⟩
                      300 `\let\MT@requires@luatex\@secondoftwo`
                      301 `\ifx\directlua\@undefined \else`
                      302   `\ifx\directlua\relax \else`
                      303   `\let\MT@requires@luatex\@firstoftwo`
                      304   `\fi`
                      305 `\fi`
                      306 ⟨debug⟩`\MT@dinfo@nl0{this is \MT@requires@luatex{}{not }luatex}`

\MT@lua   Communicate with lua. Beginning with luaTeX 0.36, \directlua no longer requires a state number. \luatexversion ought to have been enabled by the format.

                      307 `\MT@requires@luatex{`
                      308 `\ifnum\luatexversion<36`
                      309   `\def\MT@lua{\directlua0}`
                      310 `\else`
                      311   `\def\MT@lua{\directlua}`
                      312 `\fi`

Some functions are loaded from a dedicated lua file. This avoids character escaping problems and incompatibilities between versions of luaTeX. If available, we'll use the luatextra package to load the module.

                      313 `\MT@lua{`
                      314   `if (luatextra and luatextra.use_module) then`
                      315   `luatextra.use_module("microtype")`
                      316   `else`
                      317   `dofile(kpse.find_file("microtype.lua"))`

```
318    end}
319 }\relax
320 ⟨/lua⟩
321 ⟨/package⟩
322 ⟨/package | letterspace⟩
```

Here it begins. The module was contributed by Élie Roux.

```
323 ⟨*luafile⟩
324 if microtype then
325   -- we simply don't load
326 else
327
328 microtype = {}
329
330 microtype.module = {
331   name        = "microtype",
332   version     = 2.4,
333   date        = "2010/01/10",
334   description = "microtype module.",
335   author      = "R Schlicht",
336   copyright   = "R Schlicht",
337   license     = "LPPL",
338 }
339
340 if luatextra and luatextra.provides_module then
341   luatextra.provides_module(microtype.module)
342 end
343
344 ⟨/luafile⟩
```

To be continued, but first back to primitives.

\MT@glet Here's the forgotten one.

```
345 ⟨*package | letterspace⟩
346 \def\MT@glet{\global\let}
```

\MT@exp@cs   Commands to create command sequences. Those that are going to be defined
\MT@exp@gcs globally should be created inside a group so that the save stack won't explode.

```
347 \def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
348 ⟨*package⟩
349 \def\MT@exp@gcs#1#2{\begingroup\expandafter\endgroup\expandafter#1\csname#2\endcsname}
```

\MT@def@n   This is \@namedef and global.
\MT@gdef@n 350 \def\MT@def@n{\MT@exp@cs\def}
          351 \def\MT@gdef@n{\MT@exp@gcs\gdef}

\MT@edef@n   Its expanding versions.
\MT@xdef@n 352 ⟨/package⟩
          353 \def\MT@edef@n{\MT@exp@cs\edef}
          354 ⟨*package⟩
          355 \def\MT@xdef@n{\MT@exp@gcs\xdef}

\MT@let@nc   \let a \csname sequence to a command.
\MT@glet@nc 356 \def\MT@let@nc{\MT@exp@cs\let}
          357 \def\MT@glet@nc{\MT@exp@gcs\MT@glet}

\MT@let@cn   \let a command to a \csname sequence.
```
358 \def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}
```

\MT@let@nn   \let a \csname sequence to a \csname sequence.
\MT@glet@nn 359 \def\MT@let@nn{\MT@exp@cs\MT@let@cn}

360 `\def\MT@glet@nn{\MT@exp@gcs{\global\expandafter\MT@let@cn}}`

\MT@@font   Remove trailing space from the font name.

361 `\def\MT@@font{\expandafter\string\MT@font}`

\MT@exp@one@n   Expand the second token once and enclose it in braces.

362 `⟨/package⟩`
363 `\def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}`

\MT@exp@two@c   Expand the next two tokens after ⟨#1⟩ once.

364 `\def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}`
365 `⟨*package⟩`

\MT@exp@two@n   Expand the next two tokens after ⟨#1⟩ once and enclose them in braces.

366 `\def\MT@exp@two@n#1#2#3{%`
367 `  \expandafter\expandafter\expandafter`
368 `    #1\expandafter\expandafter\expandafter`
369 `      {\expandafter#2\expandafter}\expandafter{#3}}`

You do not wonder why `\MT@exp@one@c` doesn't exist, do you?

\MT@ifdefined@c@T   Wrapper for testing whether command resp. `\csname` sequence is defined. If we
\MT@ifdefined@c@TF   are running e-TeX, we will use its primitives `\ifdefined` and `\ifcsname`, which
\MT@ifdefined@n@T   decreases memory use substantially.
\MT@ifdefined@n@TF
370 `\def\MT@ifdefined@c@T#1{%`
371 `^^X  \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi`
372 `^^Q  \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi`
373 `}`
374 `⟨/package⟩`
375 `\def\MT@ifdefined@c@TF#1{%`
376 `^^X  \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi`
377 `⟨package⟩^^Q  \ifx#1\@undefined`
378 `⟨package⟩^^Q    \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi`
379 `}`
380 `\def\MT@ifdefined@n@T#1{%`
381 `^^X  \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi`
382 `⟨package⟩^^Q  \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax`
383 `⟨package⟩^^Q    \expandafter\@gobble\else\expandafter\@firstofone\fi`
384 `}`
385 `⟨*package⟩`
386 `\def\MT@ifdefined@n@TF#1{%`
387 `^^X  \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi`
388 `^^Q  \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax`
389 `^^Q    \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi`
390 `}`

\MT@detokenize@n   Translate a macro into a token list. With e-TeX, we can use `\detokenize`. We
\MT@detokenize@c   also need to remove the last trailing space; and only the last one – therefore the
\MT@rem@last@space   fiddling (and the `\string` isn't perfect, of course).

391 `\def\MT@detokenize@n#1{%`
392 `^^X  \expandafter\MT@rem@last@space\detokenize{#1} \@nil`
393 `^^Q  \string#1%`
394 `}`
395 `\def\MT@detokenize@c#1{%`
396 `^^X  \MT@exp@one@n\MT@detokenize@n#1%`
397 `^^Q  \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil`
398 `}`
399 `\def\MT@rem@last@space#1 #2{#1%`
400 `  \ifx\@nil#2\else \space`
401 `  \expandafter\MT@rem@last@space\expandafter#2\fi`

```
402 }
```

\MT@ifempty   Test whether argument is empty.

```
403 ⟨/package⟩
404 \begingroup
405 \catcode'\%=12
406 \catcode'\&=14
407 \gdef\MT@ifempty#1{&
408   \if %#1%&
409     \expandafter\@firstoftwo
410   \else
411     \expandafter\@secondoftwo
412   \fi
413 }
414 \endgroup
415 ⟨*package⟩
```

\MT@ifint   Test whether argument is an integer, using an old trick by Mr. Arseneau, or the latest and greatest from pdfTEX or luaTEX (which also allows negative numbers, as required by the letterspace option).

```
416 \MT@requires@pdftex6{
417 ⟨*lua⟩
418   \MT@requires@luatex{
419     \def\MT@ifint#1{\csname\MT@lua{microtype.ifint([[#1]])}\endcsname}
420   }{
421 ⟨/lua⟩
422 ⟨/package⟩
423     \def\MT@ifint#1{%
424       \ifcase\pdfmatch{^-*[0-9]+ *$}{#1}\relax
425         \expandafter\@secondoftwo
426       \else
427         \expandafter\@firstoftwo
428       \fi
429     }
430 ⟨*package⟩
431 ⟨lua⟩  }
432 }{
433   \def\MT@ifint#1{%
434     \if!\ifnum9<1#1!\else?\fi
435       \expandafter\@firstoftwo
436     \else
437       \expandafter\@secondoftwo
438     \fi
439   }
440 }
441 ⟨/package⟩
442 ⟨/package | letterspace⟩
443 ⟨*luafile⟩
444 function microtype.ifint(s)
445   if string.find(s,"^-*[0-9]+ *$") then
446     tex.write("@firstoftwo")
447   else
448     tex.write("@secondoftwo")
449   end
450 end
451
452 ⟨/luafile⟩
```

\MT@ifdimen   Test whether argument is dimension (or number). (nd and nc are new Didot resp.

Cicero, added in pdfTₑX 1.30; px is a pixel.)

```
453 ⟨∗package⟩
454 \MT@requires@pdftex6{
455 ⟨∗lua⟩
456   \MT@requires@luatex{
457     \def\MT@ifdimen#1{\csname\MT@lua{microtype.ifdimen([[#1]])}\endcsname}
458   }{
459 ⟨/lua⟩
460     \def\MT@ifdimen#1{%
461       \ifcase\pdfmatch{^([0-9]+([.,][0-9]+)?|[.,][0-9]+)%
462                       (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
463         \expandafter\@secondoftwo
464       \else
465         \expandafter\@firstoftwo
466       \fi
467     }
468 ⟨lua⟩  }
469 }{
470   \def\MT@ifdimen#1{%
471     \setbox\z@=\hbox{%
472       \MT@count=1#1\relax
473       \ifnum\MT@count=\@ne
474         \aftergroup\@secondoftwo
475       \else
476         \aftergroup\@firstoftwo
477       \fi
478     }%
479   }
480 }
481 ⟨/package⟩
482 ⟨∗luafile⟩
483 function microtype.ifdimen(s)
484   if (string.find(s, "^-*[0-9]+(%a*) *$") or
485       string.find(s, "^-*[0-9]*[.,][0-9]+(%a*) *$")) then
486     tex.write("@firstoftwo")
487   else
488     tex.write("@secondoftwo")
489   end
490 end
491
492 ⟨/luafile⟩
```

\MT@ifdim   Test floating point numbers.

```
493 ⟨∗package⟩
494 \def\MT@ifdim#1#2#3{%
495   \ifdim #1\p@ #2 #3\p@
496     \expandafter\@firstoftwo
497   \else
498     \expandafter\@secondoftwo
499   \fi
500 }
```

\MT@ifstreq   Test whether two strings (fully expanded) are equal.

```
501 \MT@requires@pdftex5{
502 ⟨∗lua⟩
503   \MT@requires@luatex{
504     \def\MT@ifstreq#1#2{\csname\MT@lua{microtype.ifstreq([[#1]],[[#2]])}\endcsname}
505   }{
506 ⟨/lua⟩
507     \def\MT@ifstreq#1#2{%
```

```
508        \ifcase\pdfstrcmp{#1}{#2}\relax
509          \expandafter\@firstoftwo
510        \else
511          \expandafter\@secondoftwo
512        \fi
513      }
514 ⟨lua⟩  }
515 }{
516    \def\MT@ifstreq#1#2{%
517      \edef\MT@res@a{#1}%
518      \edef\MT@res@b{#2}%
519      \ifx\MT@res@a\MT@res@b
520        \expandafter\@firstoftwo
521      \else
522        \expandafter\@secondoftwo
523      \fi
524    }
525 }
526 ⟨/package⟩
527 ⟨*luafile⟩
528 function microtype.ifstreq(s1, s2)
529    if s1 == s2 then
530      tex.write("@firstoftwo")
531    else
532      tex.write("@secondoftwo")
533    end
534 end
535
```

And here we end the `lua` file.

```
536 end
537 ⟨/luafile⟩
```

\MT@xadd   Add item to a list.

```
538 ⟨*package⟩
539 \def\MT@xadd#1#2{%
540    \ifx#1\relax
541      \xdef#1{#2}%
542    \else
543      \xdef#1{#1#2}%
544    \fi
545 }
```

\MT@xaddb   Add item to the beginning.

```
546 \def\MT@xaddb#1#2{%
547    \ifx#1\relax
548      \xdef#1{#2}%
549    \else
550      \xdef#1{#2#1}%
551    \fi
552 }
553 ⟨/package⟩
```

\MT@map@clist@n   Run ⟨#2⟩ on all elements of the comma list ⟨#1⟩. This and the following is
\MT@map@clist@c   modelled after LaTeX3 commands.
\MT@map@clist@
\MT@clist@function
\MT@clist@break

```
554 ⟨*package | letterspace⟩
555 \def\MT@map@clist@n#1#2{%
556    \ifx\@empty#1\else
557      \def\MT@clist@function##1{#2}%
558      \MT@map@clist@#1,\@nil,\@nnil
```

```
559    \fi
560  }
561  \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
562  \def\MT@map@clist@#1,{%
563    \ifx\@nil#1%
564      \expandafter\MT@clist@break
565    \fi
566    \MT@clist@function{#1}%
567    \MT@map@clist@
568  }
569  \let\MT@clist@function\@gobble
570  \def\MT@clist@break#1\@nnil{}
571  ⟨*package⟩
```

\MT@map@tlist@n   Execute ⟨#2⟩ on all elements of the token list ⟨#1⟩. \MT@tlist@break can be
\MT@map@tlist@c   used to jump out of the loop.
\MT@map@tlist@
\MT@tlist@break
```
572  \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
573  \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
574  \def\MT@map@tlist@#1#2{%
575    \ifx\@nnil#2\else
576      #1{#2}%
577      \expandafter\MT@map@tlist@
578      \expandafter#1%
579    \fi
580  }
581  \def\MT@tlist@break#1\@nnil{\fi}
```

\ifMT@inlist@   Test whether item ⟨#1⟩ is in comma list ⟨#2⟩. Using \pdfmatch would be slower.
\MT@in@clist
```
582  \newif\ifMT@inlist@
583  \def\MT@in@clist#1#2{%
584    \def\MT@res@a##1,#1,##2##3\@nnil{%
585      \ifx##2\@empty
586        \MT@inlist@false
587      \else
588        \MT@inlist@true
589      \fi
590    }%
591    \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
592  }
```

\MT@rem@from@clist   Remove item ⟨#1⟩ from comma list ⟨#2⟩. This is basically \@removeelement from
                     ltcntrl.dtx. Using \pdfmatch and \pdflastmatch here would be really slow!
```
593  \def\MT@rem@from@clist#1#2{%
594    \def\MT@res@a##1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
595    \def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%
596    \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
597  }
```

\MT@in@tlist   Test whether item is in token list. Since this isn't too elegant, I thought that at
\MT@in@tlist@   least here, \pdfmatch would be more efficient – however, it turned out to be even
                    slower than this solution.
```
598  \def\MT@in@tlist#1#2{%
599    \MT@inlist@false
600    \def\MT@res@a{#1}%
601    \MT@map@tlist@c#2\MT@in@tlist@
602  }
603  \def\MT@in@tlist@#1{%
604    \edef\MT@res@b{#1}%
605    \ifx\MT@res@a\MT@res@b
```

```
606       \MT@inlist@true
607         \expandafter\MT@tlist@break
608     \fi
609 }
```

\MT@in@rlist   Test whether size \MT@size is in a list of ranges. Store the name of the list in
\MT@in@rlist@   \MT@size@name
\MT@in@rlist@@
\MT@size@name
```
610 \def\MT@in@rlist#1{%
611     \MT@inlist@false
612     \MT@map@tlist@c#1\MT@in@rlist@
613 }
614 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
615 \def\MT@in@rlist@@#1#2#3{%
616     \MT@ifdim{#2}=\m@ne{%
617       \MT@ifdim{#1}=\MT@size
618         \MT@inlist@true
619         \relax
620     }{%
621       \MT@ifdim\MT@size<{#1}\relax{%
622         \MT@ifdim\MT@size<{#2}%
623           \MT@inlist@true
624           \relax
625       }%
626     }%
627     \ifMT@inlist@
628       \def\MT@size@name{#3}%
629       \expandafter\MT@tlist@break
630     \fi
631 }
```

\MT@loop   This is the same as LaTeX's \loop, which we mustn't use, since this could confuse
\MT@iterate   an outer \loop in the document.
\MT@repeat
```
632 ⟨/package⟩
633 \def\MT@loop#1\MT@repeat{%
634     \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
635     \MT@iterate \let\MT@iterate\relax
636 }
637 \let\MT@repeat\fi
```

\MT@while@num   Execute ⟨#3⟩ from ⟨#1⟩ up to (excluding) ⟨#2⟩ (much faster than LaTeX's
               \@whilenum).
```
638 \def\MT@while@num#1#2#3{%
639     \@tempcnta#1\relax
640     \MT@loop #3%
641       \advance\@tempcnta \@ne
642       \ifnum\@tempcnta < #2\MT@repeat
643 }
```

\MT@do@font   Execute ⟨#1⟩ 256 times.
```
644 \def\MT@do@font{\MT@while@num\z@\@cclvi}
645 ⟨*package⟩
```

\MT@count   Increment macro ⟨#1⟩ by one. Saves using up too many counters. The e-TeX way
\MT@increment is slightly faster.
```
646 \newcount\MT@count
647 \def\MT@increment#1{%
648 ^^X  \edef#1{\number\numexpr #1 + 1\relax}%
649 ^^Q  \MT@count=#1\relax
650 ^^Q  \advance\MT@count \@ne
```

```
651 ^^Q  \edef#1{\number\MT@count}%
652 }
```

\MT@scale   Multiply and divide a counter. If we are using e-TeX, we will use its \numexpr
primitive. This has the advantage that it is less likely to run into arithmetic overflow.
The result of the division will be rounded instead of truncated. Therefore, we'll get
a different (more accurate) result in about half of the cases.

```
653 \def\MT@scale#1#2#3{%
654 ^^Q  \multiply #1 #2\relax
655    \ifnum #3 = \z@
656 ^^X    #1=\numexpr #1 * #2\relax
657    \else
658 ^^X    #1=\numexpr #1 * #2 / #3\relax
659 ^^Q    \divide #1 #3\relax
660    \fi
661 }
```

\MT@abbr@pr   Some abbreviations. Thus, we can have short command names but full-length log
\MT@abbr@ex   output.
\MT@abbr@pr@c
\MT@abbr@ex@c
\MT@abbr@pr@inh
\MT@abbr@ex@inh
\MT@abbr@nl
\MT@abbr@sp
\MT@abbr@sp@c
\MT@abbr@sp@inh
\MT@abbr@kn
\MT@abbr@kn@c
\MT@abbr@kn@inh
\MT@abbr@tr
\MT@abbr@tr@c

```
662 \def\MT@abbr@pr{protrusion}
663 \def\MT@abbr@ex{expansion}
664 \def\MT@abbr@pr@c{protrusion codes}
665 \def\MT@abbr@ex@c{expansion codes}
666 \def\MT@abbr@pr@inh{protrusion inheritance}
667 \def\MT@abbr@ex@inh{expansion inheritance}
668 \def\MT@abbr@nl{noligatures}
669 \def\MT@abbr@sp{spacing}
670 \def\MT@abbr@sp@c{interword spacing codes}
671 \def\MT@abbr@sp@inh{interword spacing inheritance}
672 \def\MT@abbr@kn{kerning}
673 \def\MT@abbr@kn@c{kerning codes}
674 \def\MT@abbr@kn@inh{kerning inheritance}
675 \def\MT@abbr@tr{tracking}
676 \def\MT@abbr@tr@c{tracking amount}
```

\MT@rbba@protrusion   These we also need the other way round.
\MT@rbba@expansion
\MT@rbba@spacing
\MT@rbba@kerning
\MT@rbba@tracking

```
677 \def\MT@rbba@protrusion{pr}
678 \def\MT@rbba@expansion{ex}
679 \def\MT@rbba@spacing{sp}
680 \def\MT@rbba@kerning{kn}
681 \def\MT@rbba@tracking{tr}
```

\MT@features   We can work on these lists to save some guards in the dtx file.
\MT@features@long

```
682 \def\MT@features{pr,ex,sp,kn,tr}
683 \def\MT@features@long{protrusion,expansion,spacing,kerning,tracking}
```

\MT@is@feature   Whenever an optional argument accepts a list of features, we can use this com-
mand to check whether a feature exists in order to prevent a rather confusing
'Missing \endcsname inserted' error message. The feature (long form) must be
in \@tempa, the type of list to ignore in ⟨#1⟩, then comes the action.

```
684 \def\MT@is@feature#1{%
685    \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
686    \ifMT@inlist@
687      \expandafter\@firstofone
688    \else
689      \MT@error{'\@tempa' is not an available micro-typographic\MessageBreak
690        feature. Ignoring #1}{Available features are: '\MT@features@long'.}%
691      \expandafter\@gobble
```

```
692    \fi
693 }
```

14.1.5   Compatibility

For the record, the following LaTeX kernel commands will be modified by microtype:

- \pickup@font

- \do@subst@correction

- \add@accent (all in section 14.2.9)

- \showhyphens (in section 14.4.5)

The wordcount package redefines the font-switching commands, which will break microtype. Since microtype doesn't have an effect on the number of words in the document anyway, we will simply disable ourselves.

```
694 \@ifl@aded{tex}{wordcount}{%
695    \MT@warning@nl{Detected the 'wordcount' utility.\MessageBreak
696       Disabling '\MT@MT', since it wouldn't work}%
697    \MT@clear@options\MT@restore@catcodes\endinput}\relax
```

\MT@setup@   The setup is deferred until the end of the preamble. This has a couple of advantages: \microtypesetup can be used to change options later on in the preamble, and fonts don't have to be set up before microtype.

```
698 ⟨/package⟩
699 ⟨plain⟩\MT@requires@latex1{
700 \let\MT@setup@\@empty
```

\MT@addto@setup   We use our private hook to have better control over the timing. This will also work with eplain, but not with miniltx alone.

```
701 \def\MT@addto@setup{\g@addto@macro\MT@setup@}
```

Don't hesitate with miniltx.

```
702 ⟨plain⟩}{\let\MT@addto@setup\@firstofone}
```

\MT@with@package@T   We almost never do anything if a package is not loaded.

```
703 \def\MT@with@package@T#1{\@ifpackageloaded{#1}\@firstofone\@gobble}
704 ⟨*package⟩
```

\MT@with@babel@and@T   LaTeX's \@ifpackagewith ignores the class options.

```
705 \def\MT@with@babel@and@T#1{%
706    \MT@ifdefined@n@T{opt@babel.\@pkgextension}{%
707       \@expandtwoargs\MT@in@clist{#1}
708          {\csname opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
709       \ifMT@inlist@\expandafter\@gobble\fi
710    }\@gobble
711 }
```

Don't load letterspace.

```
712 \MT@let@nc{ver@letterspace.sty}\@empty
```

\MT@ledmac@setup
\MT@led@unhbox@line
\MT@led@kern   The ledmac package first saves each paragraph in a box, from which it then splits off the lines one by one. This will destroy character protrusion. (There aren't any problems with the lineno package, since it takes a different approach.) — ... — After much to and fro, the situation has finally settled and there is a fix. Beginning with pdfTeX version 1.21b together with ledpatch.sty as of 2005/06/02 (v0.4), character protrusion will work at last.

Peter Wilson was so kind to provide the `\l@dunhbox@line` hook in ledmac to allow for protrusion. `\leftmarginkern` and `\rightmarginkern` are new primitives of pdfTEX 1.21b (aka. 1.30.0).

```
713 \MT@requires@pdftex5{
714   \def\MT@ledmac@setup{%
715     \ifMT@protrusion
716       \MT@ifdefined@c@TF\l@dunhbox@line{%
717         \MT@info@nl{Patching ledmac to enable character protrusion}%
718         \newdimen\MT@led@kern
719         \let\MT@led@unhbox@line\l@dunhbox@line
720         \renewcommand*{\l@dunhbox@line}[1]{%
721           \ifhbox##1%
722             \MT@led@kern=\rightmarginkern##1%
723             \kern\leftmarginkern##1%
724             \MT@led@unhbox@line##1%
725             \kern\MT@led@kern
726           \fi
727         }%
728       }{%
729         \MT@warning@nl{%
730           Character protrusion in paragraphs with line\MessageBreak
731           numbering will only work if you update ledmac}%
732       }%
733     \fi
734   }
735 }{
736   \def\MT@ledmac@setup{%
737     \ifMT@protrusion
738       \MT@warning@nl{%
739         The pdftex version you are using does not allow\MessageBreak
740         character protrusion in paragraphs with line\MessageBreak
741         numbering by the 'ledmac' package.\MessageBreak
742         Upgrade pdftex to version 1.30 or later}%
743     \fi
744   }
745 }
```

`\MT@restore@p@h` Restore meaning of `\%` and `\#`.

```
746 \def\MT@restore@p@h{\chardef\%`\% \chardef\#`\# }
```

`\MT@setupfont@hook` This hook will be executed every time a font is set up (inside a group).

In the preamble, we check for the packages each time a font is set up. Thus, it will work regardless when the packages are loaded.

Even for packages that don't activate any characters in the preamble (like babel and csquotes), we have to check here, too, in case they were loaded before microtype, and a font is loaded `\AtBeginDocument`, before microtype. (This is no longer needed, since the complete setup is now deferred until the end of the preamble. However, it is still necessary for `defersetup=false`.)

```
747 \def\MT@setupfont@hook{%
```

Spanish (and Galician and Mexican) babel modify `\%`, storing the original meaning in `\percentsign`.

```
748   \MT@if@false
749   \MT@with@babel@and@T{spanish} \MT@if@true
750   \MT@with@babel@and@T{galician}\MT@if@true
751   \MT@with@babel@and@T{mexican} \MT@if@true
752   \ifMT@if@\MT@ifdefined@c@T\percentsign{\let\%\percentsign}\fi
```

Using `\@disablequotes`, we can restore the original meaning of all characters made active by csquotes. (It would be doable for older versions, too, but we won't bother.)

```
753    \MT@with@package@T{csquotes}{%
754      \@ifpackagelater{csquotes}{2005/05/11}\@disablequotes\relax}%
```

hyperref redefines `\%` and `\#` inside a `\url`. We restore the original meanings (which we can only hope are correct). Same for tex4ht.

```
755    \MT@if@false
756    \MT@with@package@T{hyperref}\MT@if@true
757    \MT@with@package@T{tex4ht}  \MT@if@true
758    \ifMT@if@\MT@restore@p@h\fi
759  }
```

Check again at the end of the preamble.

```
760 ⟨/package⟩
761 \MT@addto@setup{%
762 ⟨*package⟩
```

Our competitor, the pdfcprot package, must not be tolerated!

```
763    \MT@with@package@T{pdfcprot}{%
764      \MT@error{Detected the 'pdfcprot' package!\MessageBreak
765              '\MT@MT' and 'pdfcprot' may not be used together}{%
766 The 'pdfcprot' package provides an interface to character protrusion.\MessageBreak
767 So does the '\MT@MT' package. Using both packages at the same\MessageBreak
768 time will almost certainly lead to undesired results. Have your choice!}%
769    }%
770    \MT@with@package@T{ledmac}\MT@ledmac@setup
```

We can clean up `\MT@setupfont@hook` now.

```
771    \let\MT@setupfont@hook\@empty
772    \MT@if@false
773    \MT@with@babel@and@T{spanish} \MT@if@true
774    \MT@with@babel@and@T{galician}\MT@if@true
775    \MT@with@babel@and@T{mexican} \MT@if@true
776    \ifMT@if@
777      \g@addto@macro\MT@setupfont@hook{%
778        \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
779    \fi
780    \MT@with@package@T{csquotes}{%
781      \@ifpackagelater{csquotes}{2005/05/11}{%
782        \g@addto@macro\MT@setupfont@hook\@disablequotes
783      }{%
784        \MT@warning@nl{%
785          Should you receive warnings about unknown slot\MessageBreak
786          numbers, try upgrading the 'csquotes' package}%
787      }%
788    }%
```

We disable microtype's additions inside hyperref's `\pdfstringdef`, which redefines lots of commands. hyperref doesn't work with plain TeX, so in that case we don't bother.

```
789    \MT@if@false
790 ⟨/package⟩
791 ⟨plain⟩  \MT@requires@latex2{
792    \MT@with@package@T{hyperref}{%
793      \pdfstringdefDisableCommands{%
794 ⟨*package⟩
795        \let\pickup@font\MT@orig@pickupfont
```

```
796        \let\textmicrotypecontext\@secondoftwo
797        \let\microtypecontext\@gobble
798 ⟨/package⟩
799        \def\lsstyle{\pdfstringdefWarn\lsstyle}%
800        \def\textls#1#{\pdfstringdefWarn\textls}%
801      }%
802 ⟨package⟩      \MT@if@true
803      }%
804 ⟨plain⟩   }\relax
805 ⟨*package⟩
806    \MT@with@package@T{tex4ht}\MT@if@true
807    \ifMT@if@\g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi
```

The listings package makes numbers and letters active,

```
808    \MT@with@package@T{listings}{%
809      \g@addto@macro\MT@cfg@catcodes{%
810        \MT@while@num{"30}{"3A}{\catcode\@tempcnta 12\relax}%
811        \MT@while@num{"41}{"5B}{\catcode\@tempcnta 11\relax}%
812        \MT@while@num{"61}{"7B}{\catcode\@tempcnta 11\relax}%
813      }%
```

... and the backslash (which would lead to problems in \MT@get@slot).

```
814      \g@addto@macro\MT@setupfont@hook{%
815        \catcode'\\\z@
```

When loaded with the extendedchar option, listings will also redefine 8-bit active characters (inputenc). Luckily, this simple redefinition will make them expand to their original definition, so that they could be used in the configuration.

```
816        \let\lst@ProcessLetter\@empty
817      }%
818    }%
```

Of course, using both soul's and microtype's letterspacing mechanisms at the same time doesn't make much sense. But soul can do more, e.g., underlining. The optional argument to \textls may not be used.

```
819 ⟨/package⟩
820 ⟨plain⟩   \MT@requires@latex2{
821    \MT@with@package@T{soul}{%
822      \soulregister\lsstyle 0%
823      \soulregister\textls  1%
824    }%
```

Under plain TEX, soul doesn't register itself the LATEX way, hence we have to use a different test in this case.

```
825 ⟨*plain⟩
826    }{\ifx\SOUL@\@undefined\else
827        \soulregister\lsstyle 0%
828        \soulregister\textls  1%
829      \fi}%
830 ⟨/plain⟩
831 ⟨*package⟩
```

Compatibility with the pinyin package (from CJK): disable microtype in \py@macron, which loads a different font for the accent. In older versions of pinyin (pre-4.6.0), \py@macron had only one argument.

```
832    \MT@with@package@T{pinyin}{%
833      \let\MT@orig@py@macron\py@macron
834      \@ifpackagelater{pinyin}{2005/08/11}{% 4.6.0
835        \def\py@macron#1#2{%
```

```
836          \let\pickup@font\MT@orig@pickupfont
837          \MT@orig@py@macron{#1}{#2}%
838          \let\pickup@font\MT@pickupfont}%
839      }{%
840        \def\py@macron#1{%
841          \let\pickup@font\MT@orig@pickupfont
842          \MT@orig@py@macron{#1}%
843          \let\pickup@font\MT@pickupfont}%
844      }%
845    }%
846 ⟨/package⟩
847 }
848 ⟨*package⟩
```

We need a font (the minimal class doesn't load one).

```
849 \expandafter\ifx\the\font\nullfont\normalfont\fi
```

14.2   Font setup

\MT@setupfont   Setting up a font entails checking for each feature whether it should be applied to the current font (\MT@font). But first, we might have to disable stuff when used together with adventurous packages.

```
850 \def\MT@setupfont{\MT@setupfont@hook}
```

This will use a copy of the font (allowing for expansion parameter variation and the use of more than one set of protrusion factors for a font within one paragraph).

```
851 \MT@requires@pdftex7
852   {\g@addto@macro\MT@setupfont\MT@copy@font}\relax
```

The font properties must be extracted from \MT@font, since the current value of \f@encoding and friends may be wrong!

```
853 \g@addto@macro\MT@setupfont{%
854   \MT@exp@two@c\MT@split@name\string\MT@font/\@nil
```

Try to find a configuration file for the current font family.

```
855   \MT@exp@one@n\MT@find@file\MT@family
856   \ifx\MT@familyalias\@empty \else
857     \MT@exp@one@n\MT@find@file\MT@familyalias\fi
```

We have to make sure that \cf@encoding expands to the correct value (for later, in \MT@get@slot), which isn't the case when \selectfont chooses a new encoding (this would be done a second later in \selectfont, anyway – three lines, to be exact). (I think, I do not need this anymore – however, I'm too afraid to remove it. . . . Oops, I did it. Let's see whether anybody complains.)

```
858 % \ifx\f@encoding\cf@encoding\else\@@enc@update\fi
859 }
```

Tracking has to come first, since it means actually loading a different font.

```
860 \MT@requires@pdftex6
861   {\g@addto@macro\MT@setupfont\MT@tracking}\relax
862 \g@addto@macro\MT@setupfont{%
863   \MT@check@font
864   \ifMT@inlist@
865 ⟨debug⟩\MT@show@pdfannot2%
866   \else
867     \MT@vinfo{Setting up font '\MT@@font'\on@line}%
```

Now we can begin setting up the font for all features that the current pdfTeX provides. The following commands are `\let` to `\relax` if the respective feature is disabled via package options.

For versions older than 1.20, protrusion has to be set up first, beginning with 1.20, the order doesn't matter.

```
868    \MT@protrusion
869    \MT@expansion
870 }
```

Interword spacing and kerning (pdfTeX 1.40).

```
871 \MT@requires@pdftex6
872   {\g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}}\relax
```

Disable ligatures (pdfTeX 1.30).

```
873 \MT@requires@pdftex5
874   {\g@addto@macro\MT@setupfont\MT@noligatures}\relax
875 \g@addto@macro\MT@setupfont{%
```

Debugging.

```
876 ⟨debug⟩\MT@show@pdfannot1%
```

Finally, register the font so that we don't set it up anew each time.

```
877     \MT@register@font
878   \fi
879 }
```

`\MT@copy@font`
`\MT@copy@font@`   The new (1.40.4) `\pdfcopyfont` command allows to expand a font with different parameters, or to use more than one set of protrusion factors for a given font within one paragraph. It will be used when we find a context for `\SetProtrusion` or `\SetExpansion` in the preamble, or when the package has been loaded with the `copyfonts` option.

```
880 \let\MT@copy@font\relax
881 \MT@requires@pdftex7{
882 \def\MT@copy@font@{%
```

`\MT@font@copy`   For every new protrusion and expansion contexts, we create a new copy.

```
883   \xdef\MT@font@copy{\csname\MT@@font/\MT@pr@context/\MT@ex@context\endcsname}%
```

`\MT@font@orig`   pdfTeX doesn't allow to copy a font that has already been copied and expanded/ letterspaced. Hence, we have to get the original.

```
884   \expandafter\ifx\MT@font@copy\relax
885     \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
886     \expandafter\ifx\MT@font@orig\relax
887       \MT@exp@two@c\MT@glet\MT@font@orig\font@name
888     \else
889       \MT@exp@two@c\let\font@name\MT@font@orig
890     \fi
891     \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
892 ⟨debug⟩\MT@dinfo1{creating new copy: \MT@font@copy}%
```

Since it's a new font, we have to remove it from the context lists.

```
893     \MT@map@clist@c\MT@active@features{%
894       \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
895         \def\@tempa{##1}%
896         \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@rem@from@list
897       \fi
898     }%
899   \fi
```

```
900    \MT@exp@two@c\let\MT@font\MT@font@copy
```

We only need the font identifier for letterspacing.

```
901    \let\font@name\MT@font@copy
```

But we have to properly substitute the font after we're done.

```
902    \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
903 }
```

\MT@rem@from@list

```
904 \def\MT@rem@from@list#1{%
905   \MT@exp@cs\ifx{MT@\@tempa @#1font@list}\relax\else
906     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
907        \MT@font \csname MT@\@tempa @#1font@list\endcsname
908   \fi
909 }
910 }\relax
```

Here's the promised dirty trick   for users of older pdfTeX versions, which works around the problem that the use of the same font with different expansion parameters is prohibited. If you do not want to create a clone of the font setup (this would require duplicating the `tfm`/`vf` files under a new name, and writing new `fd` files and `map` entries), you can load a minimally larger font for the paragraph in question. E. g., for a document typeset in 10 pt:

```
\SetExpansion
   [ stretch = 30,
     shrink  = 60,
     step    =  5 ]
   { encoding = *,
     size = 10.001 }
   { }
\newcommand{\expandpar}[1]{{%
   \fontsize{10.001}{\baselineskip}\selectfont #1\par}}
% ...
\expandpar{This paragraph contains an `unnecessary' widow.}
```

Note that the \expandpar command can only be applied to complete paragraphs. If you are using Computer Modern Roman, you have to load the fix-cm package to be able to select fonts in arbitrary sizes. Finally, the reason I suggest to use a larger font, and not a smaller one, is to prevent a different design size being selected.

\MT@split@name
\MT@encoding
\MT@family
\MT@series
\MT@shape
\MT@size

Split up the font name (⟨#6⟩ may be a protrusion/expansion context and/or a letterspacing amount).

```
911 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
912   \def\MT@encoding{#1}%
913   \def\MT@family  {#2}%
914   \def\MT@series  {#3}%
915   \def\MT@shape   {#4}%
916   \def\MT@size    {#5}%
```

\MT@familyalias

Alias family?

```
917   \MT@ifdefined@n@TF{MT@\MT@family @alias}%
918     {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
919     {\let\MT@familyalias\@empty}%
920 }
```

\ifMT@do   We check all features of the current font against the lists of the currently active
\MT@feat   font set, and set \ifMT@do accordingly.

\MT@maybe@do 921 \newif\ifMT@do
              922 \def\MT@maybe@do#1{%

                  (but only if the feature isn't globally set to false)

              923   \csname ifMT@\csname MT@abbr@#1\endcsname\endcsname

                  Begin with setting micro-typography to true for this font. The \MT@checklist@...
                  tests will set it to false if the property is not in the list. The first non-empty list
                  that does not contain a match will stop us (except for font).

              924   \MT@dotrue
              925   \edef\@tempa{\csname MT@#1@setname\endcsname}%
              926   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
              927   \MT@ifdefined@n@TF{MT@checklist@##1}%
              928   {\csname MT@checklist@##1\endcsname}%
              929   {\MT@checklist@{##1}}%
              930   {#1}%
              931   }%
              932   \else
              933   \MT@dofalse
              934   \fi
              935   \ifMT@do

                  \MT@feat stores the current feature.

              936   \def\MT@feat{#1}%
              937   \csname MT@set@#1@codes\endcsname
              938   \else
              939   \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
              940   \fi
              941 }

\MT@dinfo@list

              942 ⟨debug⟩\def\MT@dinfo@list#1#2#3{\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: #2
              943 ⟨debug⟩ \ifx\\#3\\list empty\else '\@nameuse{MT@#2}' #3 list\fi}}

\MT@checklist@   The generic test (⟨#1⟩ is the axis, ⟨#2⟩ the feature, \@tempa contains the set
                  name).

              944 \def\MT@checklist@#1#2{%
              945 ⟨!debug⟩ \MT@ifdefined@n@T
              946 ⟨debug⟩   \MT@ifdefined@n@TF
              947   {MT@#2list@#1@\@tempa}{%

                  Begin a (masqueraded) \expandafter orgy to test whether the font attribute is in
                  the list.

              948   \expandafter\MT@exp@one@n\expandafter\MT@in@clist
              949   \csname MT@#1\expandafter\endcsname
              950   \csname MT@#2list@#1@\@tempa\endcsname
              951   \ifMT@inlist@
              952 ⟨debug⟩\MT@dinfo@list{#2}{#1}{in}%
              953   \MT@dotrue
              954   \else
              955 ⟨debug⟩\MT@dinfo@list{#2}{#1}{not in}%
              956   \MT@dofalse
              957   \expandafter\MT@clist@break
              958   \fi
              959   }%

                  If no limitations have been specified, i.e., the list for a font attribute has not been

defined at all, the font should be set up.

```
960 ⟨debug⟩   {\MT@dinfo@list{#2}{#1}{}}%
961 }
```

\MT@checklist@family   Also test for the alias font, if the original font is not in the list.

```
962 \def\MT@checklist@family#1{%
963 ⟨!debug⟩  \MT@ifdefined@n@T
964 ⟨debug⟩   \MT@ifdefined@n@TF
965       {MT@#1list@family@\@tempa}{%
966     \MT@exp@two@n\MT@in@clist
967         \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
968     \ifMT@inlist@
969 ⟨debug⟩\MT@dinfo@list{#1}{family}{in}%
970       \MT@dotrue
971     \else
972 ⟨debug⟩\MT@dinfo@list{#1}{family}{not in}%
973       \MT@dofalse
974     \ifx\MT@familyalias\@empty \else
975       \MT@exp@two@n\MT@in@clist
976           \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
977       \ifMT@inlist@
978 ⟨debug⟩  \MT@dinfo@list{#1}{family alias}{in}%
979           \MT@dotrue
980 ⟨debug⟩\else\MT@dinfo@list{#1}{family alias}{not in}%
981         \fi
982       \fi
983     \fi
984     \ifMT@do \else
985       \expandafter\MT@clist@break
986     \fi
987   }%
988 ⟨debug⟩   {\MT@dinfo@list{#1}{family}{}}%
989 }
```

\MT@checklist@size   Test whether font size is in list of size ranges.

```
990 \def\MT@checklist@size#1{%
991 ⟨!debug⟩  \MT@ifdefined@n@T
992 ⟨debug⟩   \MT@ifdefined@n@TF
993       {MT@#1list@size@\@tempa}{%
994     \MT@exp@cs\MT@in@rlist{MT@#1list@size@\@tempa}%
995     \ifMT@inlist@
996 ⟨debug⟩\MT@dinfo@list{#1}{size}{in}%
997       \MT@dotrue
998     \else
999 ⟨debug⟩\MT@dinfo@list{#1}{size}{not in}%
1000       \MT@dofalse
1001       \expandafter\MT@clist@break
1002     \fi
1003   }%
1004 ⟨debug⟩   {\MT@dinfo@list{#1}{size}{}}%
1005 }
```

\MT@checklist@font   If the font matches, we skip the rest of the test.

```
1006 \def\MT@checklist@font#1{%
1007 ⟨!debug⟩  \MT@ifdefined@n@T
1008 ⟨debug⟩   \MT@ifdefined@n@TF
1009       {MT@#1list@font@\@tempa}{%
```

Since \MT@font may be appended with context and/or letterspacing specs, we construct the name from the font characteristics.

```
1010        \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
1011        \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
1012          \@tempb \csname MT@#1list@font@\@tempa\endcsname
1013        \ifMT@inlist@
1014 ⟨debug⟩\MT@dinfo@list{#1}{font}{in}%
1015          \expandafter\MT@clist@break
1016        \else
1017 ⟨debug⟩\MT@dinfo@list{#1}{font}{not in}%
1018          \MT@dofalse
1019        \fi
1020    }%
1021 ⟨debug⟩   {\MT@dinfo@list{#1}{font}{}}%
1022 }
```

14.2.1 Protrusion

\MT@protrusion   Set up for protrusion?

```
1023 \def\MT@protrusion{\MT@maybe@do{pr}}
```

\MT@set@pr@codes   This macro is called by \MT@setupfont, and does all the work for setting up a font for protrusion.

```
1024 \def\MT@set@pr@codes{%
```

Check whether and if, which list should be applied to the current font.

```
1025    \MT@if@list@exists{%
1026      \MT@get@font@dimen@six{%
1027        \MT@get@opt
1028        \MT@reset@pr@codes
```

Get the name of the inheritance list and parse it.

```
1029        \MT@get@inh@list
```

Set an input encoding?

```
1030        \MT@set@inputenc{c}%
```

Load additional lists?

```
1031        \MT@load@list\MT@pr@c@name
1032        \MT@set@listname
```

Load the main list.

```
1033        \MT@let@cn\@tempc{MT@pr@c@\MT@pr@c@name}%
1034        \expandafter\MT@set@codes\@tempc,\relax,}%
1035    }\MT@reset@pr@codes
1036 }
```

\MT@get@font@dimen@six   If \fontdimen 6 is zero, character protrusion, spacing, kerning and tracking won't
\MT@dimen@six   work, and we can skip the settings (for example, the dsfont and fourier fonts don't specify this dimension; this is probably a bug in the fonts).

```
1037 \def\MT@get@font@dimen@six{%
1038    \ifnum\fontdimen6\MT@font=\z@
1039      \MT@warning@nl{%
1040        Font '\MT@@font' does not specify its\MessageBreak
1041        \@backslashchar fontdimen 6 (width of an 'em')! Therefore,\MessageBreak
1042        \@nameuse{MT@abbr@\MT@feat} will not work with this font}%
1043      \expandafter\@gobble
1044    \else
1045      \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
1046      \expandafter\@firstofone
1047    \fi
```

```
1048 }
```

\MT@set@all@pr   Set all protrusion codes of the font.

```
1049 \def\MT@set@all@pr#1#2{%
1050 ⟨debug⟩\MT@dinfo@nl{3}{-- lp/rp: setting all to #1/#2}%
1051   \let\MT@temp\@empty
1052   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lpcode\MT@font\@tempcnta=#1\relax}}%
1053   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rpcode\MT@font\@tempcnta=#2\relax}}%
1054   \MT@do@font\MT@temp
1055 }
```

\MT@reset@pr@codes@   All protrusion codes are zero for new fonts. However, if we have to reload the font
\MT@reset@pr@codes   due to different contexts, we have to reset them. This command will be changed
                       by \microtypecontext if necessary.

```
1056 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
1057 \let\MT@reset@pr@codes\relax
```

\MT@the@pr@code   If the font is letterspaced, we have to add half the letterspacing amount to the
\MT@the@pr@code@tr   margin kerns. This will be activated in \MT@set@tr@codes.

```
1058 \def\MT@the@pr@code{\@tempcntb}
1059 \MT@requires@pdftex6{
1060   \def\MT@the@pr@code@tr{%
1061     \numexpr\@tempcntb+\MT@letterspace@/2\relax
1062   }
1063 }\relax
```

\MT@set@codes   Split up the values and set the codes.

```
1064 \def\MT@set@codes#1,{%
1065   \ifx\relax#1\@empty\else
1066     \MT@split@codes #1==\relax
1067     \expandafter\MT@set@codes
1068   \fi
1069 }
```

\MT@split@codes   The keyval package would remove spaces here, which we needn't do since \SetProtrusion
                   ignores spaces in the protrusion list anyway. \MT@get@char@unit may mean diffe-
                   rent things.

```
1070 \def\MT@split@codes#1=#2=#3\relax{%
1071   \def\@tempa{#1}%
1072   \ifx\@tempa\@empty \else
1073     \MT@get@slot
1074     \ifnum\MT@char > \m@ne
1075       \MT@get@char@unit
1076       \csname MT@\MT@feat @split@val\endcsname#2\relax
1077     \fi
1078   \fi
1079 }
```

\MT@pr@split@val

```
1080 \def\MT@pr@split@val#1,#2\relax{%
1081   \def\@tempb{#1}%
1082   \MT@ifempty\@tempb\relax{%
1083     \MT@scale@to@em
1084     \lpcode\MT@font\MT@char=\MT@the@pr@code
1085 ⟨debug⟩\MT@dinfo@nl{4}{;;; lp (\MT@char): \number\lpcode\MT@font\MT@char: [#1]}%
1086   }%
1087   \def\@tempb{#2}%
1088   \MT@ifempty\@tempb\relax{%
1089     \MT@scale@to@em
```

```
1090        \rpcode\MT@font\MT@char=\MT@the@pr@code
1091 ⟨debug⟩\MT@dinfo@nl{4}{;;; rp (\MT@char): \number\rpcode\MT@font\MT@char: [#2]}%
1092      }%
```

Now we can set the values for the inheriting characters. Their slot numbers are saved in the macro \MT@inh@⟨list name⟩@⟨slot number⟩@.

```
1093   \MT@ifdefined@c@T\MT@pr@inh@name{%
1094     \MT@ifdefined@n@T{MT@inh@\MT@pr@inh@name @\MT@char @}{%
1095       \MT@exp@cs\MT@map@tlist@c
1096         {MT@inh@\MT@pr@inh@name @\MT@char @}%
1097         \MT@set@pr@heirs
1098     }%
1099   }%
1100 }
```

\MT@scale@to@em   Since pdfTEX version 0.14h, we have to adjust the protrusion factors (i. e., convert numbers from thousandths of character width to thousandths of an em of the font). We have to do this *before* setting the inheriting characters, so that the latter inherit the absolute value, not the relative one if they have a differing width (e. g., the 'ff' ligature). Unlike protcode.tex and pdfcprot, we do not calculate with \lpcode resp. \rpcode, since this would disallow protrusion factors larger than the character width (since \[lr]pcode's limit is 1000). Now, the maximum protrusion is 1 em of the font.

The unit is in \MT@count, the desired factor in \@tempb, and the result will be returned in \@tempcntb.

```
1101 \MT@requires@pdftex3{
1102   \def\MT@scale@to@em{%
1103     \@tempcntb=\MT@count\relax
```

For really huge fonts (100 pt or so), an arithmetic overflow could occur with vanilla TEX. Using e-TEX, this can't happen, since the intermediate value is 64 bit, which could only be reached with a character width larger than \maxdimen.

```
1104     \MT@scale\@tempcntb \@tempb \MT@dimen@six
1105     \ifnum\@tempcntb=\z@ \else
1106       \MT@scale@factor
1107     \fi
1108   }
```

\MT@get@charwd   Get the width of the character. When using e-TEX, we can employ \fontcharwd instead of building scratch boxes.

```
1109   \def\MT@get@charwd{%
1110 ^^X    \MT@count=\fontcharwd\MT@font\MT@char\relax
1111 ^^Q    \setbox\z@=\hbox{\MT@font \char\MT@char}%
1112 ^^Q    \MT@count=\wd\z@
1113     \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1114   }
```

For letterspaced fonts, we have to subtract the letterspacing amount from the characters' widths. The protrusion amounts will be adjusted in \MT@set@pr@codes. The letterspaced font is already loaded so that 1 em = \fontdimen 6.

```
1115   \MT@requires@pdftex6{
1116     \g@addto@macro\MT@get@charwd{%
1117       \MT@ifdefined@c@T\MT@letterspace@
1118         {\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1119     }
1120   }\relax
```

```
1121 }{
```

No adjustment with versions 0.14f and 0.14g.

```
1122   \def\MT@scale@to@em{%
1123     \MT@count=\@tempb\relax
1124     \ifnum\MT@count=\z@ \else
1125       \MT@scale@factor
1126     \fi
1127   }
```

We need this in \MT@warn@code@too@large (neutralised).

```
1128   \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1129 }
```

\MT@get@font@dimen   For the space unit.

```
1130 \def\MT@get@font@dimen#1{%
1131   \ifnum\fontdimen#1\MT@font=\z@
1132     \MT@warning@nl{Font '\MT@@font' does not specify its\MessageBreak
1133       \@backslashchar fontdimen #1 (it's zero)!\MessageBreak
1134       You should use a different 'unit' for \MT@curr@list@name}%
1135   \else
1136     \MT@count=\fontdimen#1\MT@font
1137   \fi
1138 }
```

\MT@info@missing@char   Info about missing characters, or characters with zero width.

```
1139 \def\MT@info@missing@char{%
1140   \MT@info@nl{Character '\the\MT@toks'
1141 ^^X     \iffontchar\MT@font\MT@char
1142     has a width of 0pt
1143 ^^X     \else is missing\fi
1144 ^^Q     \MessageBreak (it's probably missing)
1145     \MessageBreak in font '\MT@@font'.\MessageBreak
1146     Ignoring protrusion settings for this character}%
1147 }
```

\MT@scale@factor   Furthermore, we might have to multiply with a factor.

```
1148 \def\MT@scale@factor{%
1149   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1150     \expandafter\MT@scale\expandafter \@tempcntb
1151       \csname MT@\MT@feat @factor@\endcsname \@m
1152   \fi
1153   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
1154     \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
1155   \else
1156     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
1157       \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
1158     \fi
1159   \fi
1160 }
```

\MT@warn@code@too@large   Type out a warning if a chosen protrusion factor is too large after the conversion. As a special service, we also type out the maximum amount that may be specified in the configuration.

```
1161 \def\MT@warn@code@too@large#1{%
1162   \@tempcnta=#1\relax
1163   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1164     \expandafter\MT@scale\expandafter\@tempcnta\expandafter
1165       \@m \csname MT@\MT@feat @factor@\endcsname
1166   \fi
```

```
1167    \MT@scale\@tempcnta \MT@dimen@six \MT@count
1168    \MT@warning@nl{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space
1169      is too large for character\MessageBreak
1170      '\the\MT@toks' in \MT@curr@list@name.\MessageBreak
1171      Setting it to the maximum of \number\@tempcnta}%
1172    \@tempcntb=#1\relax
1173 }
```

\MT@get@opt   The optional argument to the configuration commands (except for \SetExpansion, which is being dealt with in \MT@get@ex@opt).

```
1174 \def\MT@get@opt{%
1175    \MT@set@listname
```

\MT@pr@factor@   Apply a factor?

\MT@sp@factor@
\MT@kn@factor@

```
1176    \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
1177       \MT@let@nn{MT@\MT@feat @factor@}
1178          {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1179       \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
1180                   \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1181    }{%
1182       \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1183    }%
```

\MT@pr@unit@   The unit can only be evaluated here, since it might be font-specific. If it's \@empty,

\MT@sp@unit@   it's relative to character widths, if it's −1, relative to space dimensions.

\MT@kn@unit@

```
1184    \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
1185       \MT@let@nn{MT@\MT@feat @unit@}%
1186          {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
1187       \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
1188          \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1189                      relative to character widths}%
1190       \else
1191          \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
1192             \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1193                         relative to width of space}%
1194          \fi
1195       \fi
1196    }{%
1197       \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
1198    }%
```

\MT@get@space@unit   The codes are either relative to character widths, or to a fixed width. For spacing

\MT@get@char@unit   and kerning lists, they may also be relative to the width of the interword glue. Only the setting from the top list will be taken into account.

```
1199    \let\MT@get@char@unit\relax
1200    \let\MT@get@space@unit\@gobble
1201    \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
1202       \let\MT@get@char@unit\MT@get@charwd
1203    \else
1204       \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
1205          \let\MT@get@space@unit\MT@get@font@dimen
1206       \else
1207          \MT@exp@cs\MT@get@unit{MT@\MT@feat @unit@}%
1208       \fi
1209    \fi
```

     Preset all characters? If so, we surely don't need to reset, too.

```
1210    \MT@ifdefined@n@T{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}{%
1211       \csname MT@preset@\MT@feat\endcsname
```

```
1212      \MT@let@nc{MT@reset@\MT@feat @codes}\relax
1213    }%
1214 }
```

\MT@get@unit
\MT@get@unit@

If `unit` contains an `em` or `ex`, we use the corresponding \fontdimen to obtain the real size. Simply converting the em into points might give a wrong result, since the font probably isn't set up yet, so that these dimensions haven't been updated, either.

```
1215 \def\MT@get@unit#1{%
1216    \expandafter\MT@get@unit@#1 e!\@nil
1217    \ifx\x\@empty\else\let#1\x\fi
1218    \@defaultunits\@tempdima#1 pt\relax\@nnil
1219    \ifdim\@tempdima=\z@
1220      \MT@warning@nl{%
1221        Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak
1222        width. Setting factors of list '\@nameuse{MT@\MT@feat @c@name}'\MessageBreak
1223        relative to character widths instead}%
1224      \let#1\@empty
1225      \let\MT@get@char@unit\MT@get@charwd
1226    \else
1227      \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} factors relative
1228                       to \the\@tempdima}%
1229      \MT@count=\@tempdima\relax
1230    \fi
1231 }
1232 \def\MT@get@unit@#1e#2#3\@nil{%
1233    \ifx\\#3\\\let\x\@empty \else
1234      \if m#2%
1235        \edef\x{#1\fontdimen6\MT@font}%
1236      \else
1237        \if x#2%
1238          \edef\x{#1\fontdimen5\MT@font}%
1239        \fi
1240      \fi
1241    \fi
1242 }
```

\MT@set@inputenc

The configurations may be under the regime of an input encoding.

```
1243 \def\MT@set@inputenc#1{%
```

\MT@cat

We remember the current category (`c` or `inh`), in case of warnings later.

```
1244    \def\MT@cat{#1}%
```

```
1245    \edef\@tempa{MT@\MT@feat @#1@\csname MT@\MT@feat @#1@name\endcsname @inputenc}%
1246    \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1247 }
```

\MT@set@inputenc@

More recent versions of inputenc remember the current encoding, so that we can test whether we really have to load the encoding file.

```
1248 \MT@addto@setup{%
1249    \@ifpackageloaded{inputenc}{%
1250      \@ifpackagelater{inputenc}{2006/02/22}{%
1251        \def\MT@set@inputenc@{%
1252          \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1253            \MT@load@inputenc
1254      }%
1255    }{%
1256      \let\MT@set@inputenc@\MT@load@inputenc
1257      }%
1258    }{%
```

```
1259     \def\MT@set@inputenc@{%
1260       \MT@warning@nl{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1261           \MessageBreak package isn't loaded. Ignoring input encoding}%
1262     }%
1263   }%
1264 }
```

\MT@load@inputenc Set up normal catcodes, since, e. g., listings would otherwise want to actually typeset the inputenc file when it is being loaded inside a listing.

```
1265 \def\MT@load@inputenc{%
1266   \MT@cfg@catcodes
1267 ⟨debug⟩\MT@dinfo@nl{1}{loading input encoding: \@nameuse{\@tempa}}%
1268   \inputencoding{\@nameuse{\@tempa}}%
1269 }
```

\MT@set@pr@heirs Set the inheriting characters.

```
1270 \def\MT@set@pr@heirs#1{%
1271   \lpcode\MT@font#1=\lpcode\MT@font\MT@char
1272   \rpcode\MT@font#1=\rpcode\MT@font\MT@char
1273 ⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
1274 ⟨debug⟩\MT@dinfo@nl{4}{;;; lp/rp (#1): \number\lpcode\MT@font\MT@char/%
1275 ⟨debug⟩                              \number\rpcode\MT@font\MT@char}%
1276 }
```

\MT@preset@pr   Preset characters. Presetting them relative to their widths is not allowed.
\MT@preset@pr@

```
1277 \def\MT@preset@pr{%
1278   \expandafter\expandafter\expandafter\MT@preset@pr@
1279     \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1280 }
1281 \def\MT@preset@pr@#1,#2\@nil{%
1282   \ifx\MT@pr@unit@\@empty
1283     \MT@warn@preset@towidth{pr}%
1284     \let\MT@preset@aux\MT@preset@aux@factor
1285   \else
1286     \def\MT@preset@aux{\MT@preset@aux@space2}%
1287   \fi
1288   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1289   \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1290   \MT@set@all@pr\@tempa\@tempb
1291 }
```

\MT@preset@aux   Auxiliary macro for presetting. Store value ⟨#1⟩ in macro ⟨#2⟩.
\MT@preset@aux@factor
\MT@preset@aux@space

```
1292 \def\MT@preset@aux@factor#1#2{%
1293   \@tempcntb=#1\relax
1294   \MT@scale@factor
1295   \edef#2{\number\@tempcntb}%
1296 }
1297 \def\MT@preset@aux@space#1#2#3{%
1298   \def\@tempb{#2}%
1299   \MT@get@space@unit#1%
1300   \MT@scale@to@em
1301   \edef#3{\number\@tempcntb}%
1302 }
```

\MT@warn@preset@towidth

```
1303 \def\MT@warn@preset@towidth#1{%
1304   \MT@warning@nl{%
1305     Cannot preset characters relative to their widths\MessageBreak
1306     for \@nameuse{MT@abbr@#1} list `\@nameuse{MT@#1@c@name}'. Presetting them%
1307     \MessageBreak relative to 1em instead}%
```

1308 `}`

14.2.2 Expansion

\MT@expansion   Set up for expansion?

1309 `\def\MT@expansion{\MT@maybe@do{ex}}`

\MT@set@ex@codes@s   Setting up font expansion is a bit different because of the `selected` option. There
are two versions of this macro.

If `selected=true`, we only apply font expansion to those fonts for which a list
has been declared (i. e., like for protrusion).

```
1310 \def\MT@set@ex@codes@s{%
1311   \MT@if@list@exists{%
1312     \MT@get@ex@opt
1313     \let\MT@get@char@unit\relax
1314     \MT@reset@ef@codes
1315     \MT@get@inh@list
1316     \MT@set@inputenc{c}%
1317     \MT@load@list\MT@ex@c@name
1318     \MT@set@listname
1319     \MT@let@cn\@tempc{MT@ex@c@\MT@ex@c@name}%
1320     \expandafter\MT@set@codes\@tempc,\relax,%
1321     \MT@expandfont
1322   }\relax
1323 }
```

\MT@set@ex@codes@n   If, on the other hand, all characters should be expanded by the same amount, we
only take the first optional argument to `\SetExpansion` into account.

\ifMT@nonselected   We need this boolean in `\MT@if@list@exists` so that no warning for missing lists
will be issued.

1324 `\newif\ifMT@nonselected`

```
1325 \def\MT@set@ex@codes@n{%
1326   \MT@nonselectedtrue
1327   \MT@if@list@exists
1328     \MT@get@ex@opt
1329   {%
1330     \let\MT@stretch@  \MT@stretch
1331     \let\MT@shrink@   \MT@shrink
1332     \let\MT@step@     \MT@step
1333     \let\MT@auto@     \MT@auto
1334     \let\MT@ex@factor@\MT@ex@factor
1335   }%
1336   \MT@reset@ef@codes
1337   \MT@expandfont
1338   \MT@nonselectedfalse
1339 }
```

\MT@set@ex@codes   Default is non-selected. It can be changed in the package options.

1340 `\let\MT@set@ex@codes\MT@set@ex@codes@n`

\MT@expandfont   Expand the font.

```
1341 \def\MT@expandfont{%
1342   \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1343 }
```

\MT@set@all@ex   At first, all expansion factors for the characters will be set to 1000 (respectively
\MT@reset@ef@codes@ the `factor` of this font).

```
1344  \def\MT@set@all@ex#1{%
1345  ⟨debug⟩\MT@dinfo@nl{3}{-- ex: setting all to \number#1}%
1346    \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1347  }
1348  \def\MT@reset@ef@codes@{\MT@set@all@ex\MT@ex@factor@}
```

\MT@reset@ef@codes   However, this is only necessary for versions prior to 1.20.

```
1349  \MT@requires@pdftex4{
1350    \def\MT@reset@ef@codes{%
1351      \ifnum\MT@ex@factor@=\@m \else
1352        \MT@reset@ef@codes@
1353      \fi
1354    }
1355  }{
1356    \let\MT@reset@ef@codes\MT@reset@ef@codes@
1357  }
```

\MT@ex@split@val   There's only one number per character.

```
1358  \def\MT@ex@split@val#1\relax{%
1359    \@tempcntb=#1\relax
```

Take an optional factor into account.

```
1360    \ifnum\MT@ex@factor@=\@m \else
1361      \MT@scale\@tempcntb \MT@ex@factor@ \@m
1362    \fi
1363    \ifnum\@tempcntb > \MT@ex@max
1364      \MT@warn@ex@too@large\MT@ex@max
1365    \else
1366      \ifnum\@tempcntb < \MT@ex@min
1367        \MT@warn@ex@too@large\MT@ex@min
1368      \fi
1369    \fi
1370    \efcode\MT@font\MT@char=\@tempcntb
1371  ⟨debug⟩\MT@dinfo@nl{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%
```

Heirs, heirs, I love thy heirs.

```
1372    \MT@ifdefined@c@T\MT@ex@inh@name{%
1373      \MT@ifdefined@n@T{MT@inh@\MT@ex@inh@name @\MT@char @}{%
1374        \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
1375      }%
1376    }%
1377  }
```

\MT@warn@ex@too@large

```
1378  \def\MT@warn@ex@too@large#1{%
1379    \MT@warning@nl{Expansion factor \number\@tempcntb\space too large for
1380      character\MessageBreak '\the\MT@toks' in \MT@curr@list@name.\MessageBreak
1381      Setting it to the maximum of \number#1}%
1382    \@tempcntb=#1\relax
1383  }
```

\MT@get@ex@opt   Apply different values to this font?
\MT@ex@factor@
\MT@stretch@
\MT@shrink@
\MT@step@
\MT@auto@

```
1384  \def\MT@get@ex@opt{%
1385    \MT@set@listname
1386    \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @factor}{%
1387      \MT@let@cn\MT@ex@factor@{MT@ex@c@\MT@ex@c@name @factor}%
1388      \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
1389    }{%
1390      \let\MT@ex@factor@\MT@ex@factor
1391    }%
1392    \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
```

```
1393    \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1394    \MT@get@ex@opt@{step}   {Setting expansion step to \number\MT@step@}%
1395    \def\@tempa{autoexpand}%
1396    \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1397    \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @preset}{%
1398      \MT@preset@ex
1399      \let\MT@reset@ef@codes\relax
1400    }%
1401  }
```

\MT@get@ex@opt@

```
1402  \def\MT@get@ex@opt@#1#2{%
1403    \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @#1}{%
1404      \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
1405      \MT@vinfo{... : #2}%
1406    }{%
1407      \MT@let@nn{MT@#1@}{MT@#1}%
1408    }%
1409  }
```

\MT@set@ex@heirs

```
1410  \def\MT@set@ex@heirs#1{%
1411    \efcode\MT@font#1=\efcode\MT@font\MT@char
1412  ⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
1413  ⟨debug⟩\MT@dinfo@nl{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1414  }
```

\MT@preset@ex

```
1415  \def\MT@preset@ex{%
1416    \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
1417    \MT@scale@factor
1418    \MT@set@all@ex\@tempcntb
1419  }
```

14.2.3 Interword spacing (glue)

\MT@spacing   Adjustment of interword spacing?

```
1420  \MT@requires@pdftex6{
1421  \def\MT@spacing{\MT@maybe@do{sp}}
```

\MT@set@sp@codes   This is all the same.

```
1422  \def\MT@set@sp@codes{%
1423    \MT@if@list@exists{%
1424      \MT@get@font@dimen@six{%
1425        \MT@get@opt
1426        \MT@reset@sp@codes
1427        \MT@get@inh@list
1428        \MT@set@inputenc{c}%
1429        \MT@load@list\MT@sp@c@name
1430        \MT@set@listname
1431        \MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}%
1432        \expandafter\MT@set@codes\@tempc,\relax,}%
1433    }\MT@reset@sp@codes
1434  }
```

\MT@sp@split@val   If unit=space, \MT@get@space@unit will be defined to fetch the corresponding
fontdimen (2 for the first, 3 for the second and 4 for the third argument).

```
1435  \def\MT@sp@split@val#1,#2,#3\relax{%
1436    \def\@tempb{#1}%
```

```
1437    \MT@ifempty\@tempb\relax{%
1438      \MT@get@space@unit2%
1439      \MT@scale@to@em
1440      \knbscode\MT@font\MT@char=\@tempcntb
1441 ⟨debug⟩\MT@dinfo@nl{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1442    }%
1443    \def\@tempb{#2}%
1444    \MT@ifempty\@tempb\relax{%
1445      \MT@get@space@unit3%
1446      \MT@scale@to@em
1447      \stbscode\MT@font\MT@char=\@tempcntb
1448 ⟨debug⟩\MT@dinfo@nl{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1449    }%
1450    \def\@tempb{#3}%
1451    \MT@ifempty\@tempb\relax{%
1452      \MT@get@space@unit4%
1453      \MT@scale@to@em
1454      \shbscode\MT@font\MT@char=\@tempcntb
1455 ⟨debug⟩\MT@dinfo@nl{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1456    }%
1457    \MT@ifdefined@c@T\MT@sp@inh@name{%
1458      \MT@ifdefined@n@T{MT@inh@\MT@sp@inh@name @\MT@char @}{%
1459        \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@sp@inh@name @\MT@char @}\MT@set@sp@heirs
1460      }%
1461    }%
1462 }
```

\MT@set@sp@heirs

```
1463 \def\MT@set@sp@heirs#1{%
1464   \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1465   \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1466   \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1467 ⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
1468 ⟨debug⟩\MT@dinfo@nl{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1469 ⟨debug⟩          \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1470 }
```

\MT@set@all@sp
\MT@reset@sp@codes
\MT@reset@sp@codes@

```
1471 \def\MT@set@all@sp#1#2#3{%
1472 ⟨debug⟩\MT@dinfo@nl{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1473   \let\MT@temp\@empty
1474   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
1475   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1476   \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1477   \MT@do@font\MT@temp
1478 }
1479 \def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
1480 \let\MT@reset@sp@codes\relax
```

\MT@preset@sp
\MT@preset@sp@

```
1481 \def\MT@preset@sp{%
1482   \expandafter\expandafter\expandafter\MT@preset@sp@
1483     \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1484 }
1485 \def\MT@preset@sp@#1,#2,#3\@nil{%
1486   \ifx\MT@sp@unit@\@empty
1487     \MT@warn@preset@towidth{sp}%
1488   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
1489   \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
1490   \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
```

```
1491    \else
1492      \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
1493      \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
1494      \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
1495    \fi
1496    \MT@set@all@sp\@tempa\@tempc\@tempb
1497 }
1498 }\relax
```

14.2.4 Additional kerning

\MT@kerning   Again, only check for additional kerning for new versions of pdfTEX.

```
1499 \MT@requires@pdftex6{
1500 \def\MT@kerning{\MT@maybe@do{kn}}
```

\MT@set@kn@codes   It's getting boring, I know.

```
1501 \def\MT@set@kn@codes{%
1502    \MT@if@list@exists{%
1503      \MT@get@font@dimen@six{%
1504        \MT@get@opt
1505        \MT@reset@kn@codes
1506        \MT@get@inh@list
1507        \MT@set@inputenc{c}%
1508        \MT@load@list\MT@kn@c@name
1509        \MT@set@listname
1510        \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
1511        \expandafter\MT@set@codes\@tempc,\relax,}%
1512    }\MT@reset@kn@codes
1513 }
```

\MT@kn@split@val   Again, the unit may be measured in the space dimension; this time only \fontdimen 2.

```
1514 \def\MT@kn@split@val#1,#2\relax{%
1515    \def\@tempb{#1}%
1516    \MT@ifempty\@tempb\relax{%
1517      \MT@get@space@unit2%
1518      \MT@scale@to@em
1519      \knbccode\MT@font\MT@char=\@tempcntb
1520 ⟨debug⟩\MT@dinfo@nl{4}{;;; knbc (\MT@char): \number\knbccode\MT@font\MT@char: [#1]}%
1521    }%
1522    \def\@tempb{#2}%
1523    \MT@ifempty\@tempb\relax{%
1524      \MT@get@space@unit2%
1525      \MT@scale@to@em
1526      \knaccode\MT@font\MT@char=\@tempcntb
1527 ⟨debug⟩\MT@dinfo@nl{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
1528    }%
1529    \MT@ifdefined@c@T\MT@kn@inh@name{%
1530      \MT@ifdefined@n@T{MT@inh@\MT@kn@inh@name @\MT@char @}{%
1531        \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
1532      }%
1533    }%
1534 }
```

\MT@set@kn@heirs

```
1535 \def\MT@set@kn@heirs#1{%
1536    \knbccode\MT@font#1=\knbccode\MT@font\MT@char
1537    \knaccode\MT@font#1=\knaccode\MT@font\MT@char
1538 ⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
1539 ⟨debug⟩\MT@dinfo@nl{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char/%
```

```
1540 ⟨debug⟩                                      \number\knaccode\MT@font\MT@char}%
1541 }
```

\MT@set@all@kn

\MT@reset@kn@codes
\MT@reset@kn@codes@

```
1542 \def\MT@set@all@kn#1#2{%
1543 ⟨debug⟩\MT@dinfo@nl{3}{-- knac/knbc: setting all to #1/#2}%
1544   \let\MT@temp\@empty
1545   \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbccode\MT@font\@tempcnta=#1\relax}}%
1546   \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knaccode\MT@font\@tempcnta=#2\relax}}%
1547   \MT@do@font\MT@temp
1548 }
1549 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1550 \let\MT@reset@kn@codes\relax
```

\MT@preset@kn
\MT@preset@kn@

```
1551 \def\MT@preset@kn{%
1552   \expandafter\expandafter\expandafter\MT@preset@kn@
1553     \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1554 }
1555 \def\MT@preset@kn@#1,#2\@nil{%
1556   \ifx\MT@kn@unit@\@empty
1557     \MT@warn@preset@towidth{kn}%
1558     \let\MT@preset@aux\MT@preset@aux@factor
1559   \else
1560     \def\MT@preset@aux{\MT@preset@aux@space2}%
1561   \fi
1562   \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1563   \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1564   \MT@set@all@kn\@tempa\@tempb
1565 }
1566 }\relax
```

14.2.5 Tracking

This only works with pdfTeX 1.40.

```
1567 \MT@requires@pdftex6{
```

\MT@tracking
\MT@tracking@

We only check whether a font should not be letterspaced at all, not whether we've already done that (because we have to do it again).

\MT@tr@font@list
```
1568 \let\MT@tr@font@list\@empty
1569 \def\MT@tracking@{%
1570   \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
1571   \ifMT@inlist@\else
1572     \MT@maybe@do{tr}%
1573     \ifMT@do\else
1574       \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1575     \fi
1576   \fi
1577 }
1578 ⟨/package⟩
1579 \let\MT@tracking
1580 ⟨package⟩  \MT@tracking@
1581 ⟨letterspace⟩  \relax
```

\MT@set@tr@codes   The tracking amount is determined by the optional argument to \textls, settings from \SetTracking, or the global letterspace option, in this order.

```
1582 \def\MT@set@tr@codes{%
1583 ⟨*package⟩
1584   \MT@vinfo{Tracking font '\MT@@font'\on@line}%
```

```
1585    \MT@get@font@dimen@six{%
1586    \MT@if@list@exists
1587      \MT@get@tr@opt
1588      \relax
1589 ⟨/package⟩
1590    \MT@ifdefined@c@TF\MT@letterspace@\relax{\let\MT@letterspace@\MT@letterspace}%
1591    \ifnum\MT@letterspace@=\z@
```

Zero tracking requires special treatment.

```
1592      \MT@set@tr@zero
1593    \else
1594 ⟨package⟩    \MT@vinfo{... Tracking by \number\MT@letterspace@}%
```

Letterspacing only works in PDF mode.

```
1595      \MT@warn@tracking@DVI
```

\MT@lsfont   The letterspaced font instances are saved in macros \⟨*font name*⟩/⟨*letterspacing amount*⟩`ls`.

In contrast to \MT@font, which may reflect the font characteristics more accurately (taking substitutions into account), \font@name is guaranteed to correspond to an actual font identifier.

```
1596    \xdef\MT@lsfont{\csname\expandafter\string\font@name
1597                    /\number\MT@letterspace@ ls\endcsname}%
1598    \expandafter\ifx\MT@lsfont\relax
1599 ⟨debug⟩\MT@dinfo@nl{1}{... new letterspacing instance}%
```

In case of nested letterspacing with different amounts, we have to extract the base font again.

```
1600      \MT@get@ls@basefont
1601      \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@
```

Scale interword spacing (not configurable in letterspace).

```
1602 ⟨*package⟩
1603      \MT@ifdefined@c@TF\MT@tr@ispace
1604        {\let\@tempa\MT@tr@ispace}%
1605        {\edef\@tempa{\MT@letterspace@*,,}}%
1606      \MT@ifdefined@c@TF\MT@tr@ospace
1607        {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1608        {\edef\@tempa{\@tempa,,,}}%
1609      \expandafter\MT@tr@set@space\@tempa,%
1610 ⟨/package⟩
1611 ⟨*letterspace⟩
1612      % spacing = {<letterspace amount>*,,}
1613      \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
1614                                    * \fontdimen2\MT@lsfont/1000\relax
1615 ⟨/letterspace⟩
```

Adjust outer kerning (microtype only).

```
1616 ⟨*package⟩
1617      \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
1618      \expandafter\MT@tr@set@okern\@tempa,%
```

Disable ligatures (not configurable in letterspace).

```
1619      \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
1620 ⟨/package⟩
1621 ⟨*letterspace⟩
1622      % no ligatures = {f}
1623      \tagcode\MT@lsfont`f=\m@ne
1624 ⟨/letterspace⟩
```

Adjust protrusion values now, and maybe later (in `\MT@pr@split@val`).

```
1625 ⟨debug⟩\MT@dinfo@nl{2}{... compensating for tracking (\number\MT@letterspace@)}%
1626         \MT@do@font{\lpcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax
1627                     \rpcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
1628 ⟨package⟩        \let\MT@the@pr@code\MT@the@pr@code@tr
1629       \fi
```

Finally, let the letterspaced font propagate.

```
1630         \aftergroup\MT@set@lsfont
1631 ⟨package⟩    \let\MT@font\MT@lsfont
```

`\MT@set@curr@ls`   We need to remember the current letterspacing amount (for `\lslig`).

`\MT@curr@ls`
```
1632       \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
1633       \aftergroup\MT@set@curr@ls
```

Adjust surrounding spacing and kerning.

`\MT@set@curr@os`   We get the current outer spacing and adjust it, then, after the end of the current outer group, set the current outer spacing, again, and adjust.

```
1634 ⟨*package⟩
1635       \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
1636       \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
1637       \MT@tr@outer@l
1638 ⟨/package⟩
```

If `\MT@ls@adjust` is empty, it's the starred version of `\textls`. Use scaling to avoid a 'Dimension too large'.

```
1639       \ifx\MT@ls@adjust\@empty
1640 ⟨letterspace⟩      % \textls : outer kerning = {*,*} ; \textls* : outer kerning = {0,0}
1641         \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
1642         \MT@ls@outer@k
1643 ⟨*letterspace⟩
1644         \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1645         \aftergroup\aftergroup\aftergroup\MT@ls@aftergroup
1646 ⟨/letterspace⟩
```

Otherwise, get the current outer kerning and adjust it, for left and right side (microtype only).

```
1647 ⟨*package⟩
1648       \else
1649         \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
1650                       \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1651         \ifdim\MT@outer@kern=\z@\else \MT@ls@outer@k \fi
1652         \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
1653                       \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
1654 ⟨/package⟩
1655       \fi
1656 ⟨*package⟩
```

`\MT@set@curr@ok`   Carry the outer kerning amount to outside the next group, then set outer spacing (which will set kerning, if no space follows).

```
1657       \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1658       \aftergroup\aftergroup\aftergroup\MT@ls@aftergroup
1659 ⟨/package⟩
1660   \fi
1661 ⟨package⟩  }%
1662 }
```

`\MT@ls@aftergroup`   Stuff to be done after the letterspace group. The letterspace package only adjusts

the kerning.

1663 ⟨letterspace⟩\def\MT@ls@aftergroup{\MT@set@curr@ok\MT@ls@outer@k}

microtype also adjusts spacing. If \tikz@expandcount is greater than zero, we're inside or at the end of a tikz node, where we don't want to do anything, lest we disturb tikz.

1664 ⟨*package⟩
1665 \MT@addto@setup{%
1666   \@ifpackageloaded{tikz}
1667   {\def\MT@ls@aftergroup{%
1668   \ifnum\tikz@expandcount>\z@ \else
1669   \MT@set@curr@os\MT@set@curr@ok\expandafter\MT@tr@outer@r\fi}}
1670   {\def\MT@ls@aftergroup{\MT@set@curr@os\MT@set@curr@ok\MT@tr@outer@r}}}

\MT@get@tr@opt   Various settings (only for the microtype version).

1671 \def\MT@get@tr@opt{%
1672   \MT@set@listname
1673   \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name}{%
1674   \MT@let@cn\MT@letterspace{MT@tr@c@\MT@tr@c@name}%

\MT@tr@unit@   Different unit?

1675   \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name @unit}{%
1676   \MT@let@cn\MT@tr@unit@{MT@tr@c@\MT@tr@c@name @unit}%
1677   \ifdim\MT@tr@unit@=1em
1678   \let\MT@tr@unit@\@undefined
1679   \else
1680   \MT@let@cn\@tempb{MT@tr@c@\MT@tr@c@name}%
1681   \MT@get@unit\MT@tr@unit@
1682   \let\MT@tr@factor@\@m
1683   \MT@scale@to@em
1684   \edef\MT@letterspace{\number\@tempcntb}%
1685   \fi
1686   }%
1687   }%

\MT@tr@ispace   Adjust interword spacing.
\MT@tr@ospace1688   \MT@get@tr@opt@{spacing}   {ispace}%
1689   \MT@get@tr@opt@{outerspacing}{ospace}%

\MT@tr@okern   Adjust outer kerning.

1690   \MT@get@tr@opt@{outerkerning}{okern}%

\MT@tr@ligatures   Which ligatures should we disable (empty means all, undefined none)?

1691   \MT@get@tr@opt@{noligatures} {ligatures}%
1692 }

\MT@get@tr@opt@

1693 \def\MT@get@tr@opt@#1#2{%
1694   \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name @#1}%
1695   {\MT@let@nn{MT@tr@#2}{MT@tr@c@\MT@tr@c@name @#1}}%
1696 }
1697 ⟨/package⟩

\MT@set@lsfont   Redefine \font@name, which will be called a second later (in \selectfont).

1698 ⟨plain⟩\MT@requires@latex2{
1699 \def\MT@set@lsfont{\MT@exp@two@c\let\font@name\MT@lsfont}

\lsstyle   Disable the tests whether the font should be letterspaced, then trigger the setup. Only \textls can be used in math mode (\lsstyle may be used inside another

text switch, of course).

```
1700 \DeclareRobustCommand\lsstyle{%
1701   \not@math@alphabet\lsstyle\textls
1702 ⟨package⟩   \def\MT@feat{tr}%
1703   \let\MT@tracking\MT@set@tr@codes
1704   \selectfont
1705 }
```

Now the definitions for the letterspace package with plain TeX.

```
1706 ⟨*plain⟩
1707 }{
1708 \def\MT@set@lsfont{\MT@lsfont}
1709 \def\lsstyle{%
1710   \begingroup
1711   \escapechar\m@ne
1712   \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1713   \MT@set@tr@codes
1714   \endgroup
1715 }
1716 \let\textls\@undefined
1717 \let\lslig\@undefined
1718 }
1719 ⟨/plain⟩
```

\lslig   For Fraktur fonts, some ligatures shouldn't be broken up. This command will
\MT@lslig   temporarily select the base font and insert the correct kerning.

```
1720 \DeclareRobustCommand\lslig[1]{%
1721   {\MT@ifdefined@c@TF\MT@curr@ls{%
1722     \escapechar\m@ne
1723     \MT@get@ls@basefont
1724     \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
1725     \kern\MT@outer@kern
1726     \font@name #1%
1727     \kern\MT@outer@kern%
1728   }{#1}}%
1729 }
```

\MT@ls@basefont   pdfTeX cannot letterspace fonts that already are letterspaced. Therefore, we have
\MT@get@ls@basefont   to save the base font in \⟨font name⟩@base.

The previous solution (checking the macro's meaning with \pdfmatch), where we were loading the base font via the \font primitive again, would destroy all previously set up micro-typographic features of the font.

```
1730 \def\MT@get@ls@basefont{%
1731   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1732   \expandafter\ifx\MT@ls@basefont\relax
1733     \MT@exp@two@c\MT@glet\MT@ls@basefont\font@name
1734   \else
1735 ⟨debug⟩\MT@dinfo@nl{1}{... fixing base font}%
1736     \MT@exp@two@c\let\font@name\MT@ls@basefont
1737   \fi
1738 }
```

\MT@set@lsbasefont   If tracking is switched off in the middle of the document, or if \textls is called
\MT@set@tr@zero   with a zero letterspacing amount, we have to retrieve the base font and select it.

```
1739 \def\MT@set@lsbasefont{\MT@exp@two@c\let\font@name\MT@ls@basefont}
1740 \def\MT@set@tr@zero{%
1741 ⟨debug⟩\MT@dinfo@nl{1}{... zero tracking}%
1742   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
```

```
1743    \expandafter\ifx\MT@ls@basefont\relax \else
1744 ⟨debug⟩\MT@dinfo@nl{1}{... fixing base font}%
1745      \aftergroup\MT@set@lsbasefont
1746    \fi
1747 }
```

\MT@tr@noligatures   pdfTeX 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.

```
1748 ⟨*package⟩
1749 \MT@requires@pdftex7{
1750    \def\MT@tr@noligatures{%
1751      \ifx\MT@tr@ligatures\@empty
1752        \MT@noligatures@\MT@lsfont\@undefined
1753      \else
1754        \MT@noligatures@\MT@lsfont\MT@tr@ligatures
1755      \fi
1756    }
1757 }{
1758    \def\MT@tr@noligatures{%
1759      \MT@warning@nl{%
1760        Disabling selected ligatures is only possible since\MessageBreak
1761        pdftex 1.40.4. Disabling all ligatures instead}%
1762      \MT@glet\MT@tr@noligatures\relax
1763    }
1764 }
```

\MT@outer@space   A new skip for outer spacing.

```
1765 \newskip\MT@outer@space
```

\MT@tr@set@space   Adjust interword spacing (\fontdimen 2–4) for inner and outer space. For inner
spacing, the font dimensions will be adjusted, the settings for outer spacing will be
remembered in a macro.

```
1766 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
1767 ⟨debug⟩\MT@dinfo@nl2{... orig. space: \the\fontdimen2\MT@lsfont,
1768 ⟨debug⟩    \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
1769 ⟨debug⟩    \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
1770    \let\MT@temp\@empty
1771    \MT@tr@set@space@{#1}{#4}{2}\@empty
1772    \MT@tr@set@space@{#2}{#5}{3}\@plus
1773    \MT@tr@set@space@{#3}{#6}{4}\@minus
1774    \MT@glet@nc{MT@outer@space\expandafter\string\font@name}\MT@temp
1775 ⟨debug⟩\MT@dinfo@nl2{... inner space: \the\fontdimen2\MT@lsfont,
1776 ⟨debug⟩    \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
1777 ⟨debug⟩\MT@dinfo@nl2{... outer space: \MT@temp}%
1778 }
```

\MT@tr@set@space@   If outer spacing settings don't exist, they will be inherited from the inner spacing
settings.

```
1779 \def\MT@tr@set@space@#1#2#3#4{%
1780    \MT@ifempty{#2}{%
1781      \MT@ifempty{#1}{%
1782        \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
1783      }{%
1784        \MT@tr@set@space@@{#1}{#3}{1000}%
1785        \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1786        \fontdimen#3\MT@lsfont=\@tempdima
1787      }%
1788    }{%
1789      \MT@tr@set@space@@{#2}{#3}{2000}%
1790      \edef\MT@temp{\MT@temp#4\the\@tempdima}%
```

```
1791        \MT@ifempty{#1}\relax{%
1792          \MT@tr@set@space@@{#1}{#3}{1000}%
1793          \fontdimen#3\MT@lsfont=\@tempdima
1794        }%
1795      }%
1796    }
```

\MT@tr@set@space@@   If the value is followed by an asterisk, the fontdimen will be scaled by the respective
                      amount, otherwise the value denotes the desired dimension in the respective unit.

```
1797    \def\MT@tr@set@space@@#1#2#3{%
1798      \MT@test@ast#1*\@nil{%
1799        \MT@ifdefined@c@TF\MT@tr@unit@
1800          {\edef\@tempb{#1}\MT@scale@to@em}
1801          {\@tempcntb=#1\relax}%
1802        \@tempdima=\dimexpr \dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
1803                            -\fontdimen#2\MT@lsfont\relax
```

For \fontdimen 2, we also have to subtract the kerning that letterspacing adds to
the sides of the characters (only half if it's for outer spacing).

```
1804        \ifnum#2=\tw@
1805          \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
1806        \fi
1807        \@tempdima=\dimexpr \fontdimen#2\MT@lsfont+\@tempdima\relax
1808      }{%
1809        \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
1810        \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
1811      }%
1812    ⟨debug⟩\MT@dinfo@nl3{... : font dimen #2 (#1): \the\@tempdima}%
1813    }
```

\MT@tr@outer@l   Recall the last skip (must really be an interword space, not just a marker, nor a
                  'hard' space, i.e., one that doesn't contain stretch or shrink parts).

```
1814    \def\MT@tr@outer@l{%
1815      \ifhmode
1816        \ifdim\lastskip>5sp
1817          \edef\x{\the\lastskip minus 0pt}%
1818          \setbox\z@\hbox{\MT@outer@space=\x}%
1819          \ifdim\wd\z@>\z@
1820    ⟨debug⟩\MT@dinfo2{[[[ adjusting pre space: \the\MT@outer@space}%
1821          \unskip \hskip\MT@outer@space\relax
```

Disable left outer kerning.

```
1822          \let\MT@ls@outer@k\relax
1823        \else
```

The ragged2e package sets \spaceskip without glue.

```
1824        \ifdim\lastskip=%
1825          \ifnum\spacefactor<2000
1826            \spaceskip
1827          \else
1828            \ifdim\xspaceskip=\z@
1829              \dimexpr\spaceskip+\fontdimen7\font@name\relax
1830            \else
1831              \xspaceskip
1832            \fi
1833          \fi
1834    ⟨debug⟩\MT@dinfo2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
1835          \unskip \hskip\MT@outer@space\relax
1836          \let\MT@ls@outer@k\relax
1837        \fi
```

```
1838        \fi
1839      \fi
1840    \fi
1841 }
```

\MT@tr@outer@next
\MT@tr@outer@r
\MT@tr@outer@r@ The following is borrowed from soul. I've added the cases for italic correction, since tracking may also be triggered by text commands (e. g., \textsc).

```
1842 \def\MT@tr@outer@r{%
1843   \futurelet\MT@tr@outer@next\MT@tr@outer@r@
1844 }
1845 \def\MT@tr@outer@r@{%
1846   \def\MT@temp*{}%
```

Don't adjust in math mode. There was a tricky bug when \textls was the last command in a \mathchoice group.

```
1847   \ifmmode \else
```

A similar bug occurred when adjustment would happen inside a discretionary group, which we prevent here. This only works with e-TEX (which we know is available).

```
1848     \ifnum\currentgrouptype=10 \else
1849       \def\MT@temp*##1{\ifhmode\hskip\MT@outer@space
1850 ⟨debug⟩\MT@dinfo2{]]] adjusting post space (1): \the\MT@outer@space}%
1851       \fi}%
1852     \ifcat\egroup\noexpand\MT@tr@outer@next

1853       \ifhmode\unkern\fi\egroup
1854       \MT@set@curr@ok \MT@set@curr@os
1855       \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\MT@temp=}%
1856     \else
```

If the next token is \maybe@ic (from an enclosing text command), we gobble it, read the next one, feed it to \maybe@ic@ (via \MT@tr@outer@icr) and then call ourselves again.

```
1857       \ifx\maybe@ic\MT@tr@outer@next
1858         \MT@set@curr@ok \MT@set@curr@os
1859         \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\MT@temp=}%
1860       \else
```

If the next token is \check@icr (from an inner text command), we insert ourselves just before it. This will then call \maybe@ic again the next round (which however will always insert an italic correction, since it doesn't read beyond our group).

```
1861         \ifx\check@icr\MT@tr@outer@next
1862           \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\MT@temp=}%
1863         \else
1864           \ifx\@sptoken\MT@tr@outer@next
1865             \def\MT@temp* {\ifhmode\hskip\MT@outer@space
1866 ⟨debug⟩\MT@dinfo2{]]] adjusting post spaces (2): \the\MT@outer@space}%
1867             \fi}%
1868           \else
1869             \ifx~\MT@tr@outer@next
1870               \def\MT@temp*~{\nobreak\hskip\MT@outer@space
1871 ⟨debug⟩\MT@dinfo2{]]] adjusting post spaces (3): \the\MT@outer@space}%
1872               }%
1873             \else
1874               \ifx\ \MT@tr@outer@next \else
1875                 \ifx\space\MT@tr@outer@next \else
1876                   \ifx\@xobeysp\MT@tr@outer@next \else
```

If there's no outer spacing, there may be outer kerning.

```
1877                          \def\MT@temp*{\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k
1878 ⟨debug⟩\MT@dinfo2{--- adjusting post kern: \the\MT@outer@kern}%
1879                            \fi}%
1880                          \let\MT@tr@outer@next\relax
1881   \fi\fi\fi\fi\fi\fi\fi\fi\fi
1882   \MT@temp*%
1883 }
```

\MT@tr@outer@icr   Helper macros for the italic correction mess.

```
\MT@tr@outer@icr@1884 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
1885 \def\MT@tr@outer@icr@{%
1886   \let\@let@token= \MT@tr@outer@next
1887   \maybe@ic@
1888 }
```

For older pdfTEX versions, throw an error.

```
1889 }{
1890   \DeclareRobustCommand\lsstyle{%
1891     \MT@error{Letterspacing only works with pdftex version 1.40\MessageBreak
1892       or newer}{Upgrade pdftex, or use the 'soul' package instead.}%
1893     \MT@glet\lsstyle\relax
1894   }
1895 }
```

And for luaTEX, too.

```
1896 ⟨*lua⟩
1897 \MT@requires@luatex{
1898   \DeclareRobustCommand\lsstyle{%
1899     \MT@error{Letterspacing currently doesn't work with luatex}
1900             {Run pdftex, or use the 'soul' package instead.}%
1901     \MT@glet\lsstyle\relax
1902   }
1903 }\relax
1904 ⟨/lua⟩
1905 ⟨/package⟩
```

\textls   This command may be used like the other text commands. The starred version
\MT@ls@adjust@   removes kerning on the sides. The optional argument changes the letterspacing
                  factor.

```
1906 \DeclareRobustCommand\textls{%
1907   \@ifstar{\let\MT@ls@adjust@\MT@ls@adjust@empty\MT@textls}%
1908           {\let\MT@ls@adjust@\MT@ls@adjust@relax\MT@textls}%
1909 }
```

\MT@textls   This is now almost LATEX's \DeclareTextFontCommand, with the difference that
\MT@letterspace@ we adjust the outer spacing and kerning also for \lsstyle, while LATEX's text
                  switches don't bother about italic correction.

```
1910 \newcommand\MT@textls[2][]{%
1911   \ifmmode
1912     \nfss@text{\MT@ls@set@ls{#1}\lsstyle#2}%
1913   \else
1914     \hmode@bgroup
1915       \MT@ls@set@ls{#1}%
1916       \lsstyle #2%
1917       \expandafter
1918     \egroup
1919   \fi
1920 }
```

\MT@ls@adjust   Set current letterspacing amount and outer kerning. This has to be done inside the
\MT@ls@adjust@empty same group as the letterspacing command.
\MT@ls@adjust@relax 1921 \def\MT@ls@adjust@empty{\let\MT@ls@adjust\@empty}
\MT@ls@set@ls   1922 \def\MT@ls@adjust@relax{\let\MT@ls@adjust\relax}
                    1923 \def\MT@ls@set@ls#1{%
                    1924   \MT@ifempty{#1}%
                    1925   {\let\MT@letterspace@\@undefined}%
                    1926   {\KV@@sp@def\MT@letterspace@{#1}%
                    1927   \MT@ls@too@large\MT@letterspace@}%
                    1928   \MT@ls@adjust@
                    1929 }

\MT@ls@too@large   Test whether letterspacing amount is too large.
                    1930 \def\MT@ls@too@large#1{%
                    1931   \ifnum#1>\MT@tr@max
                    1932   \MT@warning{Maximum for option 'letterspace' is \number\MT@tr@max}%
                    1933   \let#1\MT@tr@max
                    1934   \else
                    1935   \ifnum#1<\MT@tr@min
                    1936   \MT@warning{Minimum for option 'letterspace' is \number\MT@tr@min}%
                    1937   \let#1\MT@tr@min
                    1938   \fi
                    1939   \fi
                    1940 }

\MT@outer@kern   This dimen is used for the starred version of \textls, for \lslig and for adjusted
\MT@tr@set@okern   outer kerning.
                    1941 \newdimen\MT@outer@kern
                    1942 ⟨∗package⟩
                    1943 \def\MT@tr@set@okern#1,#2,{%
                    1944   \let\MT@temp\@empty
                    1945   \MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
                    1946   \MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
                    1947   \MT@glet@nc{MT@outer@kern\expandafter\string\font@name}\MT@temp
                    1948 ⟨debug⟩\MT@dinfo@nl2{... outer kerning: (#1,#2)
                    1949 ⟨debug⟩   = \@nameuse{MT@outer@kern\expandafter\string\font@name}}%
                    1950 }

\MT@tr@set@okern@
                    1951 \def\MT@tr@set@okern@#1{%
                    1952   \MT@test@ast#1*\@nil{%
                    1953   \MT@ifdefined@c@TF\MT@tr@unit@
                    1954   {\edef\@tempb{#1}\MT@scale@to@em}
                    1955   {\@tempcntb=#1\relax}%
                    1956   \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
                    1957   }{%
                    1958   \MT@ifempty\@tempa{\let\@tempa\@m}\relax
                    1959   \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
                    1960   * \fontdimen6\MT@lsfont/2000\relax
                    1961   }%
                    1962   \advance\@tempdima -\dimexpr \MT@letterspace@ sp
                    1963   * \fontdimen6\MT@lsfont/2000\relax
                    1964   \edef\MT@temp{\MT@temp{\the\@tempdima}}%
                    1965 }
                    1966 ⟨/package⟩

\MT@ls@outer@k   Adjust outer kerning.
                    1967 \def\MT@ls@outer@k{\ifhmode\kern\MT@outer@kern\relax\fi}
                    1968 ⟨∗package⟩

14.2.6 Disabling ligatures

\MT@noligatures   The possibility to disable ligatures is a new features of pdfTeX 1.30.

```
1969 \MT@requires@pdftex5{
1970 \def\MT@noligatures{%
1971   \MT@dotrue
1972   \let\@tempa\MT@nl@setname
1973   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1974     \MT@ifdefined@n@TF{MT@checklist@##1}%
1975       {\csname MT@checklist@##1\endcsname}%
1976       {\MT@checklist@{##1}}%
1977     {nl}%
1978   }%
1979   \ifMT@do
1980     \MT@noligatures@\MT@font\MT@nl@ligatures
1981   \fi
1982 }
```

\MT@noligatures@   This is also used by \MT@set@tr@codes.

```
1983 \def\MT@noligatures@#1#2{%
1984   \MT@ifdefined@c@TF#2{%
```

Early MiKTeX versions (before 2.5.2579) didn't know \tagcode.

```
1985     \MT@ifdefined@c@TF\tagcode{%
```

No 'inputenc' key.

```
1986       \let\MT@warn@maybe@inputenc\@empty
1987       \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
1988       \MT@map@clist@c#2{%
1989         \KV@@sp@def\@tempa{##1}\MT@get@slot
1990         \ifnum\MT@char>\m@ne \tagcode#1\MT@char=\m@ne \fi}%
1991       \MT@vinfo{... Disabling ligatures for characters: #2}%
1992     }{%
1993       \pdfnoligatures#1%
1994       \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
1995         know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
1996         the font instead}%
1997     }%
1998   }{%
1999     \pdfnoligatures#1%
2000     \MT@vinfo{... Disabling ligatures}%
2001   }%
2002 }
2003 }\relax
```

14.2.7 Loading the configuration

\MT@load@list   Recurse through the lists to be loaded.

```
2004 \def\MT@load@list#1{%
2005   \edef\@tempa{#1}%
2006   \MT@let@cn\@tempb{MT@\MT@feat @c@\@tempa @load}%
2007   \MT@ifstreq\@tempa\@tempb{%
2008     \MT@error{\@nameuse{MT@abbr@\MT@feat} list `\@tempa' cannot load itself}{}%
2009   }{%
2010     \ifx\@tempb\relax \else
2011       \MT@ifdefined@n@TF{MT@\MT@feat @c@\@tempb}{%
2012         \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list `\@tempb'}%
2013         \begingroup
2014           \MT@load@list\@tempb
```

```
2015          \endgroup
2016          \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list
2017            \noexpand\MessageBreak'\@tempb'}%
2018          \MT@let@cn\@tempc{MT@\MT@feat @c@\@tempb}%
2019          \expandafter\MT@set@codes\@tempc,\relax,%
2020        }{%
2021          \MT@error{\@nameuse{MT@abbr@\MT@feat} list '\@tempb' undefined.\MessageBreak
2022                    Cannot load it from list '\@tempa'}{}%
2023        }%
2024      \fi
2025    }%
2026 }
```

`\MT@find@file`
`\MT@file@list`

Micro-typographic settings may be written into a file mt-⟨*font family*⟩.cfg.
We must also record whether we've already loaded the file.

```
2027 \let\MT@file@list\@empty
2028 \def\MT@find@file#1{%
```

Check for existence of the file only once.

```
2029    \MT@in@clist{#1}\MT@file@list
2030    \ifMT@inlist@ \else
```

Don't forget that because reading the files takes place inside a group, all commands
that may be used there have to be defined globally.

```
2031      \MT@begin@catcodes
2032        \let\MT@begin@catcodes\relax
2033        \let\MT@end@catcodes\relax
2034        \InputIfFileExists{mt-#1.cfg}{%
2035          \edef\MT@curr@file{mt-#1.cfg}%
2036          \MT@vinfo{... Loading configuration file \MT@curr@file}%
2037          \MT@xadd\MT@file@list{#1,}%
2038        }{%
2039          \MT@get@basefamily#1\@empty\@empty\@empty\@nil
2040          \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2041          \ifMT@inlist@
2042            \MT@xadd\MT@file@list{#1,}%
2043          \else
2044            \InputIfFileExists{mt-\@tempa.cfg}{%
2045              \edef\MT@curr@file{mt-\@tempa.cfg}%
2046              \MT@vinfo{... Loading configuration file \MT@curr@file}%
2047              \MT@xadd\MT@file@list{\@tempa,#1,}%
2048            }{%
2049              \MT@vinfo{... No configuration file mt-#1.cfg}%
2050              \MT@xadd\MT@file@list{#1,}%
2051            }%
2052          \fi
2053        }%
2054      \endgroup
2055    \fi
2056 }
```

`\MT@cfg@catcodes`

We have to make sure that all characters have the correct category code. Especially,
new lines and spaces should be ignored, since files might be loaded in the middle
of the document. This is basically \nfss@catcodes (from the LATEX kernel). I've
added: & (in tabulars), !, ?, ;, : (french), ,, $, _, ~, and = (Turkish babel).

OK, now all printable characters up to 127 are 'other'. We hope that letters are
always letters and numbers other. (listings makes them active, see section 14.1.5.)

We leave ^ at catcode 7, so that stuff like '^^ff' remains possible.

```
2057 \def\MT@cfg@catcodes{%
2058   \makeatletter
2059   \catcode`\^7%
2060   \catcode`\ 9%
2061   \catcode`\^^I9%
2062   \catcode`\^^M9%
2063   \catcode`\\\z@
2064   \catcode`\{\@ne
2065   \catcode`\}\tw@
2066   \catcode`\#6%
2067   \catcode`\%14%
2068   \MT@map@tlist@n
2069     {\!\"\$\&\'\(\)\*\+\,\-\.\/\:\;\<\=\>\?\[\]\_\'\|\~}%
2070     \@makeother
2071 }
```

\MT@begin@catcodes This will be used before reading the files as well as in the configuration commands \Set..., and \DeclareCharacterInheritance, so that the catcodes are also harmless when these commands are used outside the configuration files.

```
2072 \def\MT@begin@catcodes{%
2073   \begingroup
2074   \MT@cfg@catcodes
2075 }
```

\MT@end@catcodes End group if outside configuration file (otherwise relax).

```
2076 \let\MT@end@catcodes\endgroup
```

\MT@get@basefamily The family name might have a suffix e. g., for expert set (x), old style numbers (j) swash capitals (w) etc. We mustn't simply remove the last letter, as this would make for instance cms out of cmss *and* cmsy (OK, cmex will still become cme ...).

    We only work on the font name if it is longer than three characters.

```
2077 \def\MT@get@basefamily#1#2#3#4\@nil{%
2078   \ifx\@empty#4%
2079     \def\@tempa{#1#2#3}%
2080   \else
2081     \let\@tempa\@empty
2082     \edef\@tempb{#1#2#3#4}%
2083     \expandafter\MT@get@basefamily@\@tempb\@nil
2084   \fi
2085 }
```

\MT@get@basefamily@ This will only remove one suffix (the longest match), so that *combinations* of suffixes would have be to added manually (e. g., \DeclareMicrotypeVariants*{aw}). But otherwise, something like 'padx' would be truncated to 'p'.

```
2086 \def\MT@get@basefamily@#1#2\@nil{%
2087   \edef\@tempa{\@tempa#1}%
2088   \ifx\\#2\\expandafter\@gobble\else\expandafter\@firstofone\fi
2089   {\MT@in@tlist{#2}\MT@variants
2090     \ifMT@inlist@\else\MT@get@basefamily@#2\@nil\fi}%
2091 }
```

\MT@listname Try all combinations of font family, series, shape and size to get a list for the
\MT@get@listname current font.
\MT@get@listname@
```
2092 \def\MT@get@listname#1{%
2093 ⟨debug⟩\MT@dinfo@nl{1}{trying to find \@nameuse{MT@abbr@#1} list for font `\MT@@font'}%
2094   \let\MT@listname\@undefined
2095   \def\@tempb{#1}%
2096   \MT@map@tlist@c\MT@try@order\MT@get@listname@
```

Tabelle 4: Order for matching font attributes

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
Encoding	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Family	•	•	•	•	•	•	•	•	-	-	-	-	-	-	-	-
Series	•	•	•	•	-	-	-	-	•	•	•	•	-	-	-	-
Shape	•	•	-	-	•	•	-	-	•	•	-	-	•	•	-	-
Size	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```
2097 }
2098 \def\MT@get@listname@#1{%
2099   \expandafter\MT@next@listname#1%
2100   \ifx\MT@listname\@undefined \else
2101     \expandafter\MT@tlist@break
2102   \fi
2103 }
```

\MT@try@order   Beginning with version 1.7, we always check for the font size. Since the matching order has become more logical now, it can be described in words, so that we don't need table 4 in the documentation part any longer and can cast it off here.

```
2104 \def\MT@try@order{%
2105   {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
2106   {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
2107 }
```

\MT@next@listname   The current context is added to the font attributes. That is, the context must match.

```
2108 \def\MT@next@listname#1#2#3#4{%
2109   \edef\@tempa{\MT@encoding
2110               /\ifnum#1=\@ne \MT@family\fi
2111               /\ifnum#2=\@ne \MT@series\fi
2112               /\ifnum#3=\@ne \MT@shape\fi
2113               /\ifnum#4=\@ne *\fi
2114                \MT@context}%
2115 ⟨debug⟩\MT@dinfo@nl{1}{trying \@tempa}%
2116   \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
2117     \MT@next@listname@#4%
2118   }{%
```

Also try with an alias family.

```
2119     \ifnum#1=\@ne
2120       \ifx\MT@familyalias\@empty \else
2121         \edef\@tempa{\MT@encoding
2122                     /\MT@familyalias
2123                     /\ifnum#2=\@ne \MT@series\fi
2124                     /\ifnum#3=\@ne \MT@shape\fi
2125                     /\ifnum#4=\@ne *\fi
2126                      \MT@context}%
2127 ⟨debug⟩\MT@dinfo@nl{1}{(alias) \@tempa}%
2128         \MT@ifdefined@n@T{MT@\@tempb @\@tempa}{%
2129           \MT@next@listname@#4%
2130         }%
2131       \fi
2132     \fi
2133   }%
2134 }
```

\MT@next@listname@   If size is to be evaluated, do that, otherwise use the current list.

```
2135 \def\MT@next@listname@#1{%
2136   \ifnum#1=\@ne
2137     \MT@exp@cs\MT@in@rlist{MT@\@tempb @\@tempa @sizes}%
2138     \ifMT@inlist@
2139       \let\MT@listname\MT@size@name
2140     \fi
2141   \else
2142     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
2143   \fi
2144 }
```

\MT@if@list@exists

\MT@context
```
2145 \def\MT@if@list@exists{%
2146   \MT@let@cn\MT@context{MT@\MT@feat @context}%
2147   \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
2148   \MT@get@listname{\MT@feat @c}%
2149   \MT@ifdefined@c@cTF\MT@listname{%
2150     \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
2151     \ifMT@nonselected
2152       \MT@vinfo{... Applying non-selected expansion (list '\MT@listname')}%
2153     \else
2154       \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list '\MT@listname'}%
2155     \fi
2156     \@firstoftwo
2157   }{%
```

Since the name cannot be **\@empty**, this is a sound proof that no matching list exists.

```
2158     \MT@let@nc{MT@\MT@feat @c@name}\@empty
```

Don't warn if `selected=false`.

```
2159     \ifMT@nonselected
2160       \MT@vinfo{... Applying non-selected expansion (no list)}%
2161     \else
```

Tracking doesn't require a list, either.

```
2162       \MT@ifstreq\MT@feat{tr}\relax{%
2163         \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
2164           for font\MessageBreak'\MT@@font'%
2165             \ifx\MT@context\@empty\else\space(context: '\MT@context')\fi.
2166           Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
2167       }%
2168     \fi
2169     \@secondoftwo
2170   }%
2171 }
```

\MT@get@inh@list   The inheritance lists are global (no context).

\MT@context
```
2172 \def\MT@get@inh@list{%
2173   \let\MT@context\@empty
2174   \MT@get@listname{\MT@feat @inh}%
2175   \MT@ifdefined@c@cTF\MT@listname{%
2176     \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
2177 ⟨debug⟩\MT@dinfo@nl{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
2178 ⟨debug⟩                    '\MT@listname'}%
2179     \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%
```

If the list is **\@empty**, it has already been parsed.

```
2180     \ifx\@tempc\@empty \else
2181 ⟨debug⟩\MT@dinfo@nl{1}{parsing inheritance list ...}%
```

The group is only required in case an input encoding is given.

```
2182        \begingroup
2183        \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak'\MT@listname'}%
2184        \MT@set@inputenc{inh}%
2185        \expandafter\MT@inh@do\@tempc,\relax,%
2186        \MT@glet@nc{MT@\MT@feat @inh@\MT@listname}\@empty
2187        \endgroup
2188     \fi
2189   }{%
2190     \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
2191   }%
2192 }
```

14.2.8 Translating characters into slots

Get the slot number of the character in the current encoding.

\MT@get@slot   There are lots of possibilities how a character may be specified in the configuration files, which makes translating them into slot numbers quite expensive. Also, we want to have this as robust as possible, so that the user does not have to solve a sphinx's riddle if anything goes wrong.

\MT@char   The character is in \@tempa, we want its slot number in \MT@char.

\MT@char@
```
2193 \def\MT@get@slot{%
2194   \escapechar'\\
2195   \let\MT@char@\m@ne
2196   \MT@noresttrue
```

Save unexpanded string in case we need to issue a warning message.

```
2197   \MT@toks=\expandafter{\@tempa}%
```

Now, let's walk through (hopefully) all possible cases.

- It's a letter, a character or a number.

```
2198   \expandafter\MT@is@letter\@tempa\relax\relax
2199   \ifnum\MT@char@ < \z@
```

- It might be an active character, i. e., an 8-bit character defined by inputenc. If so, we will expand it here to its LICR form.

```
2200       \MT@exp@two@c\MT@is@active\string\@tempa\@nil
```

- OK, so it must be a macro. We do not allow random commands but only those defined in LATEX's idiosyncratic font encoding scheme:

  If \⟨encoding⟩\⟨command⟩ (that's *one* command) is defined, we try to extract the slot number.

  We must be cautious not to stumble over accented characters consisting of two commands, like \'\i or \U\CYRI, hence, \string wouldn't be safe enough.

```
2201       \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
2202         \MT@is@symbol
```

- Now, we'll catch the rest, which hopefully is an accented character (e. g. \"a).

```
2203        {\expandafter\MT@is@composite\@tempa\relax\relax}%
2204      \ifnum\MT@char@ < \z@
```

- It could also be a `\chardef`ed command (e. g., the percent character). This seems the least likely case, so it's last.

```
2205        \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2206          \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2207      \fi
2208    \fi

2209    \let\MT@char\MT@char@
2210    \ifnum\MT@char < \z@
2211      \MT@warn@unknown
2212    \else
```

If the user has specified something like 'fi', or wanted to define a number but forgot to use three digits, we'll have something left of the string. In this case, we issue a warning and forget the complete string.

```
2213      \ifMT@norest \else
2214        \MT@warn@rest
2215        \let\MT@char\m@ne
2216      \fi
2217    \fi
2218    \escapechar\m@ne
2219 }
```

`\ifMT@norest`   Test whether all of the string has been used up.

```
2220 \newif\ifMT@norest
```

`\MT@is@letter`   Input is a letter, a character or a number.

```
2221 \def\MT@is@letter#1#2\relax{%
2222   \ifcat a\noexpand#1\relax
2223     \edef\MT@char@{\number'#1}%
2224     \ifx\\#2\\%
2225 ⟨debug⟩\MT@dinfo@nl{3}{> '\the\MT@toks' is a letter (\MT@char@)}%
2226     \else
2227       \MT@norestfalse
2228     \fi
2229   \else
2230     \ifcat !\noexpand#1\relax
2231       \edef\MT@char@{\number'#1}%
2232 ⟨debug⟩\MT@dinfo@nl{3}{> '\the\MT@toks' is a character (\MT@char@)}%
2233       \ifx\\#2\\%
2234         \ifnum\MT@char@ > 127 \MT@warn@ascii \fi
2235       \else
2236         \MT@norestfalse
2237         \expandafter\MT@is@number#1#2\relax\relax
2238       \fi
2239     \fi
2240   \fi
2241 }
```

`\MT@is@number`   Numbers may be specified as a three-digit decimal number (029), as a hexadecimal number (prefixed with ": "1D) or as a octal number (prefixed with ': '35). They must consist of at least three characters (including the prefix), that is, "F is not permitted.

```
2242 \def\MT@is@number#1#2#3\relax{%
2243   \ifx\relax#3\relax \else
```

```
2244      \ifx\relax#2\relax \else
2245        \MT@noresttrue
2246        \if#1"\relax
2247          \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2248 ⟨debug⟩\MT@dinfo@nl{3}{> ... a hexadecimal number: \MT@char@}%
2249        \else
2250          \if#1'\relax
2251            \def\MT@char@{\number#1#2#3}%
2252 ⟨debug⟩\MT@dinfo@nl{3}{> ... an octal number: \MT@char@}%
2253          \else
2254            \MT@ifint{#1#2#3}{%
2255              \def\MT@char@{\number#1#2#3}%
2256 ⟨debug⟩\MT@dinfo@nl{3}{> ... a decimal number: \MT@char@}%
2257            }\MT@norestfalse
2258          \fi
2259        \fi
2260      \ifnum\MT@char@ > \@cclv
2261        \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2262        \let\MT@char@\m@ne
2263      \fi
2264    \fi
2265  \fi
2266 }
```

\MT@is@active   Expand an active character. (This was completely broken in v1.7, and only worked
by chance before.) We \set@display@protect to translate, e. g., Ä into \"A, that
is to whatever it is defined in the inputenc encoding file.

Unfortunately, the (older) inputenc definitions prefer the protected/generic
variants (e. g., \copyright instead of \textcopyright), which our parser won't
be able to understand. (I'm fed up now, so you have to complain if you really,
really want to be able to write '©' instead of \textcopyright, thus rendering your
configuration files unportable.)

Unicode characters (inputenc/utf8,utf8x) are also supported.

```
2267 \def\MT@is@active#1#2\@nil{%
2268  \ifnum\catcode'#1 = \active
2269    \begingroup
2270      \set@display@protect
2271      \let\IeC\@firstofone
2272      \let\@inpenc@undefined@\MT@undefined@char
```

We refrain from checking whether there is a sufficient number of octets.

```
2273      \def\UTFviii@defined##1{\ifx ##1\relax
2274        \MT@undefined@char{utf8}\else\expandafter ##1\fi}%
```

For ucs (utf8x). Let's call it experimental . . .

```
2275      \MT@ifdefined@c@T\PrerenderUnicode
2276        {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
2277      \edef\x{\endgroup
2278        \def\noexpand\@tempa{\@tempa}%
```

Append what we think the translation is to the token register we use for the log.

```
2279      \MT@toks={\the\MT@toks\space(= \@tempa)}%
2280    }%
2281    \x
2282  \fi
2283 }
```

\MT@undefined@char   For characters not defined in the current input encoding.

2284 `\def\MT@undefined@char#1{undefined in input encoding ``#1''}`

`\MT@is@symbol` The symbol commands might expand to funny stuff, depending on context. Instead of simply expanding \⟨*command*⟩, we construct the command \⟨*encoding*⟩\⟨*command*⟩ and see whether its meaning is `\char"`⟨*hex number*⟩, which is the case for everything that has been defined with `\DeclareTextSymbol` in the encoding definition files.

```
2285 \def\MT@is@symbol{%
2286   \expandafter\def\expandafter\MT@char\expandafter
2287     {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2288   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2289     \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2290   \ifnum\MT@char@ < \z@
```

... or, if it hasn't been defined by `\DeclareTextSymbol`, a letter (e. g., `\i`, when using frenchpro).

```
2291     \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2292   \fi
2293 }
```

`\MT@is@char` A helper macro that inspects the `\meaning` of its argument.

`\MT@charstring`
```
2294 \begingroup
2295   \catcode`\/=\z@
2296   /MT@map@tlist@n{/\CHAR}/@makeother
2297   /lowercase{%
2298     /def/x{/endgroup
2299       /def/MT@charstring{\CHAR"}%
2300       /def/MT@is@char##1\CHAR"##2##3##4/relax{%
2301         /ifx/relax##1/relax
2302           /if##3\/relax
2303             /edef/MT@char@{/number"##2}%
2304             /MT@ifstreq/MT@charstring{##3##4}/relax/MT@norestfalse
2305           /else
2306             /edef/MT@char@{/number"##2##3}%
2307             /MT@ifstreq/MT@charstring{##4}/relax/MT@norestfalse
2308           /fi
2309 ⟨debug⟩   /MT@dinfo@nl{3}{> `/the/MT@toks' is a \char (/MT@char@)}%
2310         /fi
2311       }%
2312     }%
2313   }
2314 /x
```

`\MT@is@composite` Here, we are dealing with accented characters, specified as two tokens.

```
2315 \def\MT@is@composite#1#2\relax{%
2316   \ifx\\#2\\\else
```

Again, we construct a control sequence, this time of the form: \\⟨*encoding*⟩ \⟨*accent*⟩-⟨*character*⟩, e. g., `\\T1\"-a`, which we then expand once to see if it is a letter (if it has been defined by `\DeclareTextComposite`). This should be robust, finally, especially, since we also `\detokenize` the input instead of only `\string`ifying it. Thus, we will die gracefully even on wrong Unicode input without utf8.

```
2317     \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2318                     \string\csname\MT@encoding\endcsname
2319                     \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
2320     \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
```

```
2321    \fi
2322 }
```

[What about math? Well, for a moment the following looked like a solution, with \mt@is@mathchar defined accordingly, analogous to \MT@is@char above, to pick up the last two tokens (the \meaning of a \mathchardef'ed command expands to its hexadecimal notation):

```
\def\MT@is@mathchar#1{%
  \if\relax\noexpand#1% it's a macro
    \let\x#1%
  \else % it's a character
    \mathchardef\x=\mathcode`#1\relax
  \fi
  \expandafter\MT@exp@two@c\expandafter\mt@is@mathchar\expandafter
    \meaning\expandafter\x\mt@mathcharstring\relax\relax\relax
}
```

However, the problem is that \mathcodes and \mathchardefs have global scope. Therefore, if they are changed by a package that loads different math fonts, there is no guarantee whatsoever that things will still be correct (e.g., the minus in cmsy when the euler package is loaded). So, no way to go, unfortunately.]

Some warning messages, for performance reasons separated here.

\MT@curr@list@name   The type and name of the current list, defined at various places.

\MT@set@listname
```
2323 \def\MT@set@listname{%
2324   \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
2325     `\@nameuse{MT@\MT@feat @c@name}'}%
2326 }
```

\MT@warn@ascii   For 'other' characters > 127, we issue a warning (inputenc probably hasn't been loaded), since correspondence with the slot numbers would be purely coincidental.

```
2327 \def\MT@warn@ascii{%
2328   \MT@warning@nl{Character `\the\MT@toks' (= \MT@char@)
2329     is outside of ASCII range.\MessageBreak
2330     You must load the `inputenc' package before using\MessageBreak
2331     8-bit characters in \MT@curr@list@name}%
2332 }
```

\MT@warn@number@too@large   Number too large.

```
2333 \def\MT@warn@number@too@large#1{%
2334   \MT@warning@nl{%
2335     Number #1 in encoding `\MT@encoding' too large!\MessageBreak
2336     Ignoring it in \MT@curr@list@name}%
2337 }
```

\MT@warn@rest   Not all of the string has been parsed.

```
2338 \def\MT@warn@rest{%
2339   \MT@warning@nl{%
2340     Unknown slot number of character\MessageBreak`\the\MT@toks'%
2341     \MT@warn@maybe@inputenc\MessageBreak
2342     in font encoding `\MT@encoding'.\MessageBreak
2343     Make sure it's a single character\MessageBreak
2344     (or a number) in \MT@curr@list@name}%
2345 }
```

\MT@warn@unknown   No idea what went wrong.

```
2346 \def\MT@warn@unknown{%
2347   \MT@warning@nl{%
2348     Unknown slot number of character\MessageBreak`\the\MT@toks'%
```

```
2349        \MT@warn@maybe@inputenc\MessageBreak
2350        in font encoding '\MT@encoding' in \MT@curr@list@name}%
2351 }
```

\MT@warn@maybe@inputenc   In case an input encoding had been requested.

```
2352 \def\MT@warn@maybe@inputenc{%
2353   \MT@ifdefined@n@T
2354     {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}%
2355     { (input encoding '\@nameuse
2356       {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}')}%
2357 }
```

14.2.9 Hook into LaTeX's font selection

We append \MT@setupfont to \pickup@font, which is called by LaTeX every time
a font is selected. We then check whether we've already seen this font, and if not,
set it up for micro-typography. This ensures that we will catch all fonts, and that
we will not set up fonts more than once. The whole package really hangs on this
command.

In contrast to the pdfcprot package, it is not necessary to declare in advance
which fonts should benefit from micro-typographic treatment. Also, only those
fonts that are actually being used will be set up.

For my reference:

- \pickup@font is called by \selectfont, \wrong@fontshape, or
  \getanddefine@fonts (for math).

- \pickup@font calls \define@newfont.

- \define@newfont may call (inside a group!)

  - \wrong@fontshape, which in turn will call \pickup@font, and thus
    \define@newfont again, or

  - \extract@font.

- \get@external@font is called by \extract@font, by itself, and by the substitution
  macros.

Up to version 1.3 of this package, we were using \define@newfont as the hook,
which is only called for *new* fonts, and therefore seemed the natural choice. However,
this meant that we had to take special care to catch all fonts: we additionally had
to set up the default font, the error font (if it wasn't the default font), we had to
check for some packages that might have been loaded before microtype and were
loading fonts, e.g., jurabib, ledmac, pifont (loaded by hyperref), tipa, and probably
many more. Furthermore, we had to include a hack for the IEEEtran class which
loads all fonts in the class file itself (to fine tune inter-word spacing), and the
memoir class, too. To cut this short: it seemed to get out of hand, and I decided
that it would be better to use \pickup@font and decide for ourselves whether
we've already seen that font. I hope the overhead isn't too large.

\MT@font@list   We use a comma separated list.

\MT@font

```
2358 \let\MT@font@list\@empty
2359 \let\MT@font\@empty
```

All this is done at the beginning of the document. It doesn't work for plain, of course, which doesn't have `\pickup@font`.

```
2360 ⟨/package⟩
2361 ⟨plain⟩\MT@requires@latex2{
2362 \MT@addto@setup{%
```

`\MT@orig@pickupfont`   microtype also works with CJK in the sense that nothing will break when both packages are used at the same time. However, since CJK has its own way of encoding, it is currently not possible to create character-specific settings. That is, the only feature available with CJK fonts is expansion. (Tracking doesn't really work for other reasons.) Like us, CJK redefines `\pickup@font`.

```
2363   \@ifpackageloaded{CJK}{%
2364     \@ifpackagelater{CJK}{2006/10/17}% 4.7.0
2365       {\def\MT@orig@pickupfont{\CJK@ifundefined\CJK@plane}}%
2366       {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}%
2367     \g@addto@macro\MT@orig@pickupfont
2368       {{\expandafter\ifx\font@name\relax\define@newfont\fi}}%
```

CJKutf8 redefines `\pickup@font` once more (recent versions, in PDF mode, as determined by ifpdf, which CJKutf8 loads).

```
2369     \@ifpackageloaded{CJKutf8}%
2370       {\@ifpackagelater{CJKutf8}{2008/05/22}% 4.8.0
2371         {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}%
2372         {\@firstoftwo}}%
2373       {\@firstoftwo}%
2374     {\g@addto@macro\MT@orig@pickupfont{%
2375       {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2376         \define@newfont\else\xdef\font@name{%
2377           \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
2378     {\g@addto@macro\MT@orig@pickupfont{%
2379       {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2380         \define@newfont\def\CJK@temp{v}%
2381         \ifx\CJK@temp\CJK@plane
2382           \expandafter\ifx\csname CJK@cmap@\f@family\CJK@plane\endcsname\relax
2383           \else\csname CJK@cmap@\f@family\CJK@plane\endcsname\fi
2384         \else \CJK@addcmap\CJK@plane \fi
2385       \else\xdef\font@name{%
2386         \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
2387   }{%
2388     \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}%
2389   }%
```

Check whether `\pickup@font` is defined as expected. The warning issued by `\CheckCommand*` would be a bit too generic.

```
2390   \ifx\pickup@font\MT@orig@pickupfont \else
2391     \MT@warning@nl{%
2392       Command \string\pickup@font\space is not defined as expected.%
2393       \MessageBreak Patching it anyway. Some things may break%
2394 ⟨*package⟩
2395     .\MessageBreak Double-check whether micro-typography is indeed%
2396       \MessageBreak applied to the document.%
2397       \MessageBreak (Hint: Turn on 'verbose' mode)%
2398 ⟨/package⟩
2399     }%
2400   \fi
```

`\pickup@font`   Then we append our stuff. Everything is done inside a group.

```
2401   \g@addto@macro\pickup@font{\begingroup}%
```

If the trace package is loaded, we turn off tracing of microtype's setup, which is extremely noisy.

```
2402    \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
2403    \g@addto@macro\pickup@font{%
2404        \escapechar\m@ne
2405 ⟨∗package⟩
2406 ⟨debug⟩        \global\MT@inannottrue
2407 ⟨debug⟩        \MT@glet\MT@pdf@annot\@empty
2408 ⟨debug⟩        \MT@addto@annot{(line \number\inputlineno)}%
```

If \MT@font is empty, no substitution has taken place, hence \font@name is correct. Otherwise, if they are different, \font@name does not describe the font actually used. This test will catch first order substitutions, like bx to b, but it will still fail if the substituting font is itself substituted.

```
2409        \MT@let@cn\MT@font{MT@subst@\expandafter\string\font@name}%
2410        \ifx\MT@font\relax
2411          \let\MT@font\font@name
2412        \else
2413          \ifx\MT@font\font@name \else
2414 ⟨debug⟩   \MT@addto@annot{= substituted with \MT@@font}%
2415            \MT@register@subst@font
2416          \fi
2417        \fi
2418        \MT@setupfont
2419 ⟨/package⟩
2420 ⟨letterspace⟩        \MT@tracking
2421        \endgroup
2422    }%
2423 ⟨∗package⟩
```

\MT@pickupfont   Remember the patched command for later.

```
2424    \let\MT@pickupfont\pickup@font
```

\do@subst@correction   Additionally, we hook into \do@subst@correction, which is called if a substitution has taken place, to record the name of the ersatz font. Unfortunately, this will only work for one-level substitutions. We have to remember the substitute for the rest of the document, not just for the first time it is called, since we need it every time a font is letterspaced.

```
2425    \g@addto@macro\do@subst@correction
2426      {\edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
2427        \MT@glet@nc{MT@subst@\expandafter\string\font@name}\MT@font}%
```

\add@accent   Inside \add@accent, we have to disable microtype's setup, since the grouping in
\MT@orig@add@accent   the patched \pickup@font would break the accent if different fonts are used for the base character and the accent. Fortunately, LATEX takes care that the fonts used for the \accent are already set up, so that we cannot be overlooking them.

```
2428    \let\MT@orig@add@accent\add@accent
2429    \def\add@accent#1#2{%
2430      \let\pickup@font\MT@orig@pickupfont
2431      \MT@orig@add@accent{#1}{#2}%
2432      \let\pickup@font\MT@pickupfont
2433    }%
2434 ⟨/package⟩
2435 }
2436 ⟨plain⟩}\relax
2437 ⟨∗package⟩
```

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\MT@check@font Check whether we've already seen the current font.

2438 `\def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}`

\MT@register@subst@font Register the substituted font.

2439 `\def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}`

\MT@register@font Register the current font.

2440 `\def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}`

14.2.10 Context-sensitive setup

Here are the variants for context-sensitive setup.

\MT@active@features The activated features are stored in this command.

2441 `\let\MT@active@features\@empty`

\MT@check@font@cx Every feature has its own list of fonts that have already been dealt with. If the font needn't be set up for a feature, we temporarily disable the corresponding setup command. This should be more efficient than book-keeping the fonts in lists associated with the combination of contexts, as we've done it before.

```
2442 \def\MT@check@font@cx{%
2443   \MT@if@true
2444   \MT@map@clist@c\MT@active@features{%
2445     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
2446       \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2447     \ifMT@inlist@
2448       \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
2449     \else
2450       \MT@if@false
2451     \fi
2452   }%
2453   \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2454 }
```

\MT@register@subst@font@cx Add the substituted font to each feature list.

```
2455 \def\MT@register@subst@font@cx{%
2456   \MT@map@clist@c\MT@active@features{%
2457     \MT@exp@cs\MT@xadd
2458       {MT@##1@\csname MT@##1@context\endcsname font@list}%
2459       {\font@name,}%
2460   }%
2461 }
```

\MT@register@font@cx For each feature, add the current font to the list, unless we didn't set it up.

```
2462 \def\MT@register@font@cx{%
2463   \MT@map@clist@c\MT@active@features{%
2464     \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
2465       \MT@exp@cs\MT@xadd
2466         {MT@##1@\csname MT@##1@context\endcsname font@list}%
2467         {\MT@font,}%
2468       \def\@tempa{##1}%
2469       \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
2470     \fi
2471   }%
2472 }
```

\MT@maybe@rem@from@list   Recurse through all context font lists of the document and remove the font, unless
                           it's the current context.

```
2473 \def\MT@maybe@rem@from@list#1{%
2474   \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
2475     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
2476       \MT@font \csname MT@\@tempa @#1font@list\endcsname
2477   }%
2478 }
```

\microtypecontext   The user may change the context, so that different setups are possible. This is
                         especially useful for multi-lingual documents.
                             Inside the preamble, it shouldn't actually do anything but remember it for
                         later.

```
2479 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
2480 \MT@addto@setup{%
2481   \DeclareRobustCommand\microtypecontext[1]{%
2482     \MT@setup@contexts
2483     \let\MT@reset@context\relax
2484     \setkeys{MTC}{#1}%
2485     \selectfont
2486     \MT@reset@context
2487   }%
2488 }
```

\textmicrotypecontext   This is just a wrapper around \microtypecontext.

```
2489 \DeclareRobustCommand\textmicrotypecontext[2]{{\microtypecontext{#1}#2}}
```

\MT@reset@context   We have to reset the font at the end of the group, provided there actually was a
\MT@reset@context@   change.

```
2490 \def\MT@reset@context@{%
2491   \MT@vinfo{<<< Resetting contexts\on@line
2492 ⟨debug⟩    \MessageBreak= \MT@pr@context/\MT@ex@context
2493 ⟨debug⟩                  /\MT@tr@context/\MT@kn@context/\MT@sp@context
2494   }%
2495   \selectfont
2496 }
```

\MT@setup@contexts   The first time \microtypecontext is called, we initialise the context lists and
                         redefine the commands used in \pickup@font.

```
2497 \def\MT@setup@contexts{%
2498   \MT@map@clist@c\MT@active@features
2499     {\MT@glet@nc{MT@##1@@font@list}\MT@font@list}%
2500   \MT@glet\MT@check@font\MT@check@font@cx
2501   \MT@glet\MT@register@font\MT@register@font@cx
2502   \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
2503   \MT@glet\MT@setup@contexts\relax
2504 }
```

Define context keys.

```
2505 \MT@map@clist@c\MT@features@long{%
2506   \define@key{MTC}{#1}[]{%
2507     \edef\@tempb{\@nameuse{MT@rbba@#1}}%
2508     \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
2509     \ifMT@inlist@
```

Using an empty context is only asking for trouble, therefore we choose the '@'
instead (hoping for the LATEX users' natural awe of this character).

```
2510       \MT@ifempty{##1}{\def\MT@val{@}}{\def\MT@val{##1}}%
```

```
2511        \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
2512 ⟨debug⟩\MT@dinfo{1}{>>> no change of #1 context: '\MT@val'}%
2513        \else
2514          \MT@vinfo{>>> Changing #1 context to '\MT@val'\MessageBreak\on@line
2515 ⟨debug⟩            \space(previous: '\@nameuse{MT@\@tempb @context}')%
2516                }%
2517          \def\MT@reset@context{\aftergroup\MT@reset@context@}%
```

The next time we see the font, we have to reset *all* factors.

```
2518          \MT@glet@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes@}%
```

We must also keep track of all contexts in the document.

```
2519          \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
2520            \MT@val \csname MT@\@tempb @doc@contexts\endcsname
2521          \ifMT@inlist@ \else
2522            \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{{\MT@val}}%
2523 ⟨debug⟩   \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
2524          \fi
2525          \MT@edef@n{MT@\@tempb @context}{\MT@val}%
2526        \fi
2527      \fi
2528    }%
2529 }
```

<div style="float:left">
\MT@pr@context
\MT@ex@context
\MT@tr@context
\MT@sp@context
\MT@kn@context
\MT@pr@doc@contexts
\MT@ex@doc@contexts
\MT@tr@doc@contexts
\MT@sp@doc@contexts
\MT@kn@doc@contexts
\DeclareMicrotypeSet
\MT@extra@context
\DeclareMicrotypeSet*
</div>

Initialise the contexts.

```
2530 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
2531   \MT@def@n{MT@#1@context}{@}%
2532   \MT@def@n{MT@#1@doc@contexts}{{@}}%
2533 }
2534 \let\MT@extra@context\@empty
```

14.3 Configuration

14.3.1 Font sets

Calling this macro will create a comma list for every font attribute of the form: $\MT\langle feature\rangle\texttt{list@}\langle attribute\rangle\texttt{@}\langle set\ name\rangle$. If the optional argument is empty, lists for all available features will be created.

The third argument must be a list of `key=value` pairs. If a font attribute is not specified, we define the corresponding list to `\relax`, so that it does not constitute a constraint.

```
2535 \def\DeclareMicrotypeSet{%
2536   \@ifstar
2537     \MT@DeclareSetAndUseIt
2538     \MT@DeclareSet
2539 }
```

\MT@DeclareSet

```
2540 \newcommand\MT@DeclareSet[3][]{%
2541   \KV@@sp@def\@tempa{#1}%
2542   \MT@ifempty\@tempa{%
2543     \MT@map@clist@c\MT@features{{\MT@declare@sets{##1}{#2}{#3}}}%
2544   }{%
2545     \MT@map@clist@c\@tempa{{%
2546       \KV@@sp@def\@tempa{##1}%
2547       \MT@ifempty\@tempa\relax{%
2548         \MT@is@feature{set declaration '#2'}{%
2549           \MT@exp@one@n\MT@declare@sets
```

```
2550              {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
2551            }%
2552          }%
2553       }}%
2554     }%
2555 }
```

\MT@DeclareSetAndUseIt

```
2556 \newcommand\MT@DeclareSetAndUseIt[3][]{%
2557   \MT@DeclareSet[#1]{#2}{#3}%
2558   \UseMicrotypeSet[#1]{#2}%
2559 }
```

\MT@curr@set@name We need to remember the name of the set currently being declared.

```
2560 \let\MT@curr@set@name\@empty
```

\MT@declare@sets Define the current set name and parse the keys.

```
2561 \def\MT@declare@sets#1#2#3{%
2562   \KV@@sp@def\MT@curr@set@name{#2}%
2563   \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
2564     \MT@warning{Redefining \@nameuse{MT@abbr@#1} set '\MT@curr@set@name'}%
2565     \MT@glet@nc{MT@#1list@size@\MT@curr@set@name}\@empty
2566   }%
2567   \MT@glet@nc{MT@#1@set@@\MT@curr@set@name}\@empty
2568 ⟨debug⟩\MT@dinfo{1}{declaring \@nameuse{MT@abbr@#1} set '\MT@curr@set@name'}%
2569   \setkeys{MT@#1@set}{#3}%
2570 }
```

\MT@define@set@key@ ⟨#1⟩ = font axis, ⟨#2⟩ = feature.

```
2571 \def\MT@define@set@key@#1#2{%
2572   \define@key{MT@#2@set}{#1}[]{%
2573     \MT@glet@nc{MT@#2list@#1@\MT@curr@set@name}\@empty
2574     \MT@map@clist@n{##1}{%
2575       \KV@@sp@def\MT@val{####1}%
2576       \MT@get@highlevel{#1}%
```

We do not add the expanded value to the list . . .

```
2577       \MT@exp@two@n\g@addto@macro
2578         {\csname MT@#2list@#1@\MT@curr@set@name\expandafter\endcsname}%
2579         {\MT@val,}%
2580     }%
```

. . . but keep in mind that the list has to be expanded at the end of the preamble.

```
2581     \expandafter\g@addto@macro\expandafter\MT@font@sets
2582       \csname MT@#2list@#1@\MT@curr@set@name\endcsname
2583 ⟨debug⟩\MT@dinfo@nl{1}{-- #1: \@nameuse{MT@#2list@#1@\MT@curr@set@name}}%
2584   }%
2585 }
```

\MT@get@highlevel Saying, for instance, 'family=rm*' or 'shape=bf*' will expand to \rmdefault resp. \bfdefault.

```
2586 \def\MT@get@highlevel#1{%
2587   \expandafter\MT@test@ast\MT@val*\@nil\relax{%
```

And 'family = *' will become \familydefault.

```
2588     \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
2589     \edef\MT@val{\expandafter\noexpand\csname \@tempa default\endcsname}%
```

In contrast to earlier version, these values will not be expanded immediately but at the end of the preamble.

```
2590     }%
2591 }
```

\MT@test@ast   It the last character is an asterisk, execute the second argument, otherwise the
first one.

```
2592 \def\MT@test@ast#1*#2\@nil{%
2593   \def\@tempa{#1}%
2594   \MT@ifempty{#2}%
2595 }
```

\MT@font@sets   Fully expand the font specification and fix catcodes for all font sets.

```
\MT@fix@font@set 2596 \let\MT@font@sets\@empty
2597 \def\MT@fix@font@set#1{%
2598   \xdef#1{#1}%
2599   \global\@onelevel@sanitize#1%
2600 }
```

\MT@define@set@key@size   `size` requires special treatment.

```
2601 \def\MT@define@set@key@size#1{%
2602   \define@key{MT@#1@set}{size}[]{%
2603     \MT@map@clist@n{##1}{%
2604       \KV@@sp@def\MT@val{####1}%
2605       \expandafter\MT@get@range\MT@val--\@nil
2606       \ifx\MT@val\relax \else
2607         \MT@exp@cs\MT@xadd
2608           {MT@#1list@size@\MT@curr@set@name}%
2609           {{{\MT@lower}{\MT@upper}\relax}}%
2610       \fi
2611     }%
2612 ⟨debug⟩\MT@dinfo@nl{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
2613   }%
2614 }
```

Font sizes may also be specified as ranges. This has been requested by Andreas
Bühmann, who has also offered valuable help in implementing this. Now, it is
for instance possible to set up different lists for fonts with optical sizes. (The
MinionPro project is trying to do this for the OpenType version of Adobe's Minion.
See http://developer.berlios.de/projects/minionpro/.)

\MT@get@range   Ranges will be stored as triplets of {⟨lower bound⟩}{⟨upper bound⟩}{⟨list name⟩}.
\MT@upper   For simple sizes, the upper boundary is −1.
\MT@lower

```
2615 \def\MT@get@range#1-#2-#3\@nil{%
2616   \MT@ifempty{#1}{%
2617     \MT@ifempty{#2}{%
2618       \let\MT@val\relax
2619     }{%
2620       \def\MT@lower{0}%
2621       \def\MT@val{#2}%
2622       \MT@get@size
2623       \edef\MT@upper{\MT@val}%
2624     }%
2625   }{%
2626     \def\MT@val{#1}%
2627     \MT@get@size
2628     \ifx\MT@val\relax \else
2629       \edef\MT@lower{\MT@val}%
2630       \MT@ifempty{#2}{%
2631         \MT@ifempty{#3}%
2632           {\def\MT@upper{-1}}%
```

2048 pt is TeX's maximum font size.

```
2633              {\def\MT@upper{2048}}%
2634        }{%
2635          \def\MT@val{#2}%
2636          \MT@get@size
2637          \ifx\MT@val\relax \else
2638            \MT@ifdim\MT@lower>\MT@val{%
2639              \MT@error{%
2640                Invalid size range (\MT@lower\space > \MT@val) in font set
2641                '\MT@curr@set@name'.\MessageBreak Swapping sizes}{}%
2642              \edef\MT@upper{\MT@lower}%
2643              \edef\MT@lower{\MT@val}%
2644            }{%
2645              \edef\MT@upper{\MT@val}%
2646            }%
2647            \MT@ifdim\MT@lower=\MT@upper
2648              {\def\MT@upper{-1}}%
2649              \relax
2650          \fi
2651        }%
2652      \fi
2653    }%
2654 }
```

\MT@get@size   Translate a size selection command and normalise it.

```
2655 \def\MT@get@size{%
```

A single star would mean \sizedefault, which doesn't exist, so we define it to be
\normalsize.

```
2656   \if*\MT@val\relax
2657     \def\@tempa{\normalsize}%
2658   \else
2659     \MT@let@cn\@tempa{\MT@val}%
2660   \fi
2661   \ifx\@tempa\relax \else
```

The relsize solution of parsing \@setfontsize does not work with the AMS
classes, among others. I hope my hijacking doesn't do any harm. We redefine
\set@fontsize, and not \@setfontsize because some classes might define the
size selection commands by simply using \fontsize (e. g., the a0poster class).

```
2662     \begingroup
2663       \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
2664       \@tempa\@nil
2665   \fi
```

Test whether we finally got a number or dimension so that we can strip the 'pt'
(\@defaultunits and \strip@pt are kernel macros).

```
2666   \MT@ifdimen\MT@val{%
2667     \@defaultunits\@tempdima\MT@val pt\relax\@nnil
2668     \edef\MT@val{\strip@pt\@tempdima}%
2669   }{%
2670     \MT@warning{Could not parse font size '\MT@val'\MessageBreak
2671                 in font set '\MT@curr@set@name'}%
2672     \let\MT@val\relax
2673   }%
2674 }
```

\MT@define@set@key@font

```
2675 \def\MT@define@set@key@font#1{%
```

```
2676    \define@key{MT@#1@set}{font}[]{%
2677      \MT@glet@nc{MT@#1list@font@\MT@curr@set@name}\@empty
2678      \MT@map@clist@n{##1}{%
2679        \KV@@sp@def\MT@val{####1}%
2680        \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/*}}\relax
2681        \expandafter\MT@get@font\MT@val/////\@nil
2682        \MT@exp@two@n\g@addto@macro
2683          {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
2684          {\MT@val,}%
2685      }%
2686      \expandafter\g@addto@macro\expandafter\MT@font@sets
2687        \csname MT@#1list@font@\MT@curr@set@name\endcsname
2688 ⟨debug⟩\MT@dinfo@nl{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}%
2689    }%
2690 }
```

\MT@get@font   Translate any asterisks.

```
2691 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2692   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
2693   \ifx\MT@val\relax\def\MT@val{0}\fi
2694   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2695   \let\MT@val\@tempb
2696 }
```

\MT@get@font@   Helper macro, also used by \MT@get@font@and@size.

```
2697 \def\MT@get@font@#1#2#3#4#5#6{%
2698   \let\@tempb\@empty
2699   \def\MT@temp{#1/#2/#3/#4/#5}%
2700   \MT@get@axis{encoding}{#1}%
2701   \MT@get@axis{family}   {#2}%
2702   \MT@get@axis{series}   {#3}%
2703   \MT@get@axis{shape}    {#4}%
2704   \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
2705   \MT@ifempty{#5}{%
2706     \MT@warn@axis@empty{size}{\string\normalsize}%
2707     \def\MT@val{*}%
2708   }{%
2709     \def\MT@val{#5}%
2710   }%
2711   \MT@get@size
2712 }
```

\MT@get@axis

```
2713 \def\MT@get@axis#1#2{%
2714   \def\MT@val{#2}%
2715   \MT@get@highlevel{#1}%
2716   \MT@ifempty\MT@val{%
2717     \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2718     \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2719   }\relax
2720   \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
2721 }
```

\MT@warn@axis@empty

```
2722 \def\MT@warn@axis@empty#1#2{%
2723   \MT@warning{#1 axis is empty in font specification\MessageBreak
2724     '\MT@temp'. Using '#2' instead}%
2725 }
```

We can finally assemble all pieces to define \DeclareMicrotypeSet's keys. They are also used for \DisableLigatures.

```
2726 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
2727   \MT@define@set@key@{encoding}{#1}%
2728   \MT@define@set@key@{family}  {#1}%
2729   \MT@define@set@key@{series}  {#1}%
2730   \MT@define@set@key@{shape}   {#1}%
2731   \MT@define@set@key@size      {#1}%
2732   \MT@define@set@key@font      {#1}%
2733 }
```

\UseMicrotypeSet   To use a particular set we simply redefine MT@⟨feature⟩@setname. If the optional argument is empty, set names for all features will be redefined.

```
2734 \renewcommand*\UseMicrotypeSet[2][]{%
2735   \KV@@sp@def\@tempa{#1}%
2736   \MT@ifempty\@tempa{%
2737     \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
2738   }{%
2739     \MT@map@clist@c\@tempa{{%
2740       \KV@@sp@def\@tempa{##1}%
2741       \MT@ifempty\@tempa\relax{%
2742         \MT@is@feature{activation of set '#2'}{%
2743           \MT@exp@one@n\MT@use@set
2744             {\csname MT@rbba@\@tempa\endcsname}{#2}%
2745         }%
2746       }%
2747     }}%
2748   }%
2749 }
```

\MT@pr@setname   Only use sets that have been declared.
\MT@ex@setname
\MT@tr@setname
\MT@sp@setname
\MT@kn@setname
\MT@use@set

```
2750 \def\MT@use@set#1#2{%
2751   \KV@@sp@def\@tempa{#2}%
2752   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
2753     \MT@xdef@n{MT@#1@setname}{\@tempa}%
2754   }{%
2755     \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2756       \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2757     }%
2758     \MT@error{%
2759       The \@nameuse{MT@abbr@#1} set '\@tempa' is undeclared.\MessageBreak
2760       Using set '\@nameuse{MT@#1@setname}' instead}{}%
2761   }%
2762 }
```

\DeclareMicrotypeSetDefault   This command can be used in the main configuration file to declare the default font set, in case no set is specified in the package options.

```
2763 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
2764   \KV@@sp@def\@tempa{#1}%
2765   \MT@ifempty\@tempa{%
2766     \MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
2767   }{%
2768     \MT@map@clist@c\@tempa{{%
2769       \KV@@sp@def\@tempa{##1}%
2770       \MT@ifempty\@tempa\relax{%
2771         \MT@is@feature{declaration of default set '#2'}{%
2772           \MT@exp@one@n\MT@set@default@set
2773             {\csname MT@rbba@\@tempa\endcsname}{#2}%
2774         }%
```

```
2775          }%
2776        }}%
2777      }%
2778 }
```

\MT@default@pr@set

\MT@default@ex@set
```
2779 \def\MT@set@default@set#1#2{%
```
\MT@default@tr@set
```
2780   \KV@@sp@def\@tempa{#2}%
```
\MT@default@sp@set
```
2781   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
2782 ⟨debug⟩\MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set '\@tempa'}%
```
\MT@default@kn@set
```
2783     \MT@xdef@n{MT@default@#1@set}{\@tempa}%
```
\MT@set@default@set
```
2784   }{%
2785     \MT@error{%
2786       The \@nameuse{MT@abbr@#1} set '\@tempa' is not declared.\MessageBreak
2787       Cannot make it the default set. Using set\MessageBreak 'all' instead}{}%
2788     \MT@xdef@n{MT@default@#1@set}{all}%
2789   }%
2790 }
```

14.3.2 Variants and aliases

\DeclareMicrotypeVariants
\MT@variants

Specify suffixes for variants (see `fontname/variants.map`). The starred version appends to the list.

```
2791 \let\MT@variants\@empty
2792 \def\DeclareMicrotypeVariants{%
2793   \@ifstar
2794     \MT@DeclareVariants
2795     {\let\MT@variants\@empty\MT@DeclareVariants}%
2796 }
```

\MT@DeclareVariants

```
2797 \def\MT@DeclareVariants#1{%
2798   \MT@map@clist@n{#1}{%
2799     \KV@@sp@def\@tempa{##1}%
2800     \@onelevel@sanitize\@tempa
2801     \xdef\MT@variants{\MT@variants{\@tempa}}%
2802   }%
2803 }
```

\DeclareMicrotypeAlias

This can be used to set an alias name for a font, so that the file and the settings for the aliased font will be loaded.

```
2804 \renewcommand*\DeclareMicrotypeAlias[2]{%
2805   \KV@@sp@def\@tempa{#1}%
2806   \KV@@sp@def\@tempb{#2}%
2807   \@onelevel@sanitize\@tempb
2808   \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2809     \MT@warning{Alias font family '\@tempb' will override
2810       alias '\@nameuse{MT@\@tempa @alias}'\MessageBreak
2811       for font family '\@tempa'}}%
2812   \MT@xdef@n{MT@\@tempa @alias}{\@tempb}%
```

If we encounter this command while a font is being set up, we also set the alias for the current font so that if \DeclareMicrotypeAlias has been issued inside a configuration file, the configuration file for the alias font will be loaded, too.

```
2813   \MT@ifdefined@c@T\MT@family{%
2814 ⟨debug⟩\MT@dinfo{1}{Activating alias font '\@tempb' for '\MT@family'}%
2815     \MT@glet\MT@familyalias\@tempb
2816   }%
```

```
2817 }
```

\LoadMicrotypeFile   May be used to load a configuration file manually.

```
2818 \def\LoadMicrotypeFile#1{%
2819   \KV@@sp@def\@tempa{#1}%
2820   \@onelevel@sanitize\@tempa
2821   \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2822   \ifMT@inlist@
2823     \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2824   \else
2825     \MT@xadd\MT@file@list{\@tempa,}%
2826     \MT@begin@catcodes
2827     \InputIfFileExists{mt-\@tempa.cfg}{%
2828       \edef\MT@curr@file{mt-\@tempa.cfg}%
2829       \MT@vinfo{... Loading configuration file \MT@curr@file}%
2830     }{%
2831       \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2832                   does not exist}%
2833     }%
2834     \MT@end@catcodes
2835   \fi
2836 }
```

14.3.3 Disabling ligatures

\DisableLigatures   This is really simple now: we can re-use the set definitions of \DeclareMicrotypeSet;
\MT@DisableLigatures   there can only be one set, which we'll call 'no ligatures'.
\MT@nl@setname   The optional argument may be used to disable selected ligatures only.
\MT@nl@ligatures

```
2837 \MT@requires@pdftex5{
2838 \def\DisableLigatures{%
2839   \MT@begin@catcodes
2840   \MT@DisableLigatures
2841 }
2842 \newcommand*\MT@DisableLigatures[2][]{%
2843   \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
2844   \xdef\MT@active@features{\MT@active@features,nl}%
2845   \global\MT@noligaturestrue
2846   \MT@declare@sets{nl}{no ligatures}{#2}%
2847   \gdef\MT@nl@setname{no ligatures}%
2848   \MT@end@catcodes
2849 }
2850 }{
```

If pdfTEX is too old, we throw an error.

```
2851 \renewcommand*\DisableLigatures[2][]{%
2852   \MT@error{Disabling ligatures of a font is only possible\MessageBreak
2853     with pdftex version 1.30 or newer.\MessageBreak
2854     Ignoring \string\DisableLigatures}{Upgrade pdftex.}%
2855 }
2856 }
```

14.3.4 Interaction with babel

\DeclareMicrotypeBabelHook   Declare the context that should be loaded when a babel language is selected. The
command will not check whether a previous declaration will be overwritten.

```
2857 \def\DeclareMicrotypeBabelHook#1#2{%
2858   \MT@map@clist@n{#1}{%
```

```
2859       \KV@@sp@def\@tempa{##1}%
2860       \MT@gdef@n{MT@babel@\@tempa}{#2}%
2861   }%
2862 }
```

14.3.5   Fine tuning

The commands \SetExpansion and \SetProtrusion provide an interface for setting the character protrusion resp. expansion factors for a set of fonts.

\SetProtrusion   This macro accepts three arguments: [options,] set of font attributes and list of character protrusion factors.

A new macro called \MT@pr@c@⟨name⟩ will be defined to be ⟨#3⟩ (i.e., the list of characters, not expanded).

```
2863 \def\SetProtrusion{%
2864   \MT@begin@catcodes
2865   \MT@SetProtrusion
2866 }
```

\MT@SetProtrusion   We want the catcodes to be correct even if this is called in the preamble.

\MT@pr@c@name
\MT@extra@context
```
2867 \newcommand*\MT@SetProtrusion[3][]{%
2868   \let\MT@extra@context\@empty
```

\MT@permutelist   Parse the optional first argument. We first have to know the name before we can deal with the extra options.

```
2869   \MT@set@named@keys{MT@pr@c}{#1}%
2870 ⟨debug⟩\MT@dinfo{1}{creating protrusion list '\MT@pr@c@name'}%
2871   \def\MT@permutelist{pr@c}%
2872   \setkeys{MT@cfg}{#2}%
```

We have parsed the second argument, and can now define macros for all permutations of the font attributes to point to \MT@pr@c@⟨name⟩, . . .

```
2873   \MT@permute
```

. . . which we can now define to be ⟨#3⟩. Here, as elsewhere, we have to make the definitions global, since they will occur inside a group.

```
2874   \MT@gdef@n{MT@pr@c@\MT@pr@c@name}{#3}%
2875   \MT@end@catcodes
2876 }
```

\SetExpansion   \SetExpansion only differs in that it allows some extra options (stretch, shrink, step, auto).

```
2877 \def\SetExpansion{%
2878   \MT@begin@catcodes
2879   \MT@SetExpansion
2880 }
```

\MT@SetExpansion

\MT@ex@c@name
\MT@extra@context
\MT@permutelist
```
2881 \newcommand*\MT@SetExpansion[3][]{%
2882   \let\MT@extra@context\@empty
2883   \MT@set@named@keys{MT@ex@c}{#1}%
2884   \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @factor}{%
2885     \ifnum\csname MT@ex@c@\MT@ex@c@name @factor\endcsname > \@m
2886       \MT@warning@nl{Expansion factor \number\@nameuse{MT@ex@c@\MT@ex@c@name @factor}
2887         too large in list\MessageBreak '\MT@ex@c@name'. Setting it to the
2888         maximum of 1000}%
2889       \MT@glet@nc{MT@ex@c@\MT@ex@c@name @factor}\@m
2890     \fi
```

```
2891     }%
2892 ⟨debug⟩\MT@dinfo{1}{creating expansion list '\MT@ex@c@name'}%
2893     \def\MT@permutelist{ex@c}%
2894     \setkeys{MT@cfg}{#2}%
2895     \MT@permute
2896     \MT@gdef@n{MT@ex@c@\MT@ex@c@name}{#3}%
2897     \MT@end@catcodes
2898 }
```

\SetTracking

```
2899 \def\SetTracking{%
2900     \MT@begin@catcodes
2901     \MT@SetTracking
2902 }
```

\MT@SetTracking   Third argument may be empty.

```
2903 \newcommand*\MT@SetTracking[3][]{%
2904     \let\MT@extra@context\@empty
2905     \MT@set@named@keys{MT@tr@c}{#1}%
2906 ⟨debug⟩\MT@dinfo{1}{creating tracking list '\MT@tr@c@name'}%
2907     \def\MT@permutelist{tr@c}%
2908     \setkeys{MT@cfg}{#2}%
2909     \MT@permute
2910     \KV@@sp@def\@tempa{#3}%
2911     \MT@ifempty\@tempa\relax{%
2912       \MT@ifint\@tempa
2913         {\MT@xdef@n{MT@tr@c@\MT@tr@c@name}{\@tempa}}%
2914         {\MT@warning{Value '\@tempa' is not a number in\MessageBreak
2915                     tracking set '\MT@curr@set@name'}}}%
2916     \MT@end@catcodes
2917 }
```

\SetExtraSpacing

```
2918 \def\SetExtraSpacing{%
2919     \MT@begin@catcodes
2920     \MT@SetExtraSpacing
2921 }
```

\MT@SetExtraSpacing
\MT@sp@c@name
\MT@extra@context
\MT@permutelist

```
2922 \newcommand*\MT@SetExtraSpacing[3][]{%
2923     \let\MT@extra@context\@empty
2924     \MT@set@named@keys{MT@sp@c}{#1}%
2925 ⟨debug⟩\MT@dinfo{1}{creating spacing list '\MT@sp@c@name'}%
2926     \def\MT@permutelist{sp@c}%
2927     \setkeys{MT@cfg}{#2}%
2928     \MT@permute
2929     \MT@gdef@n{MT@sp@c@\MT@sp@c@name}{#3}%
2930     \MT@end@catcodes
2931 }
```

\SetExtraKerning

```
2932 \def\SetExtraKerning{%
2933     \MT@begin@catcodes
2934     \MT@SetExtraKerning
2935 }
```

\MT@SetExtraKerning
\MT@kn@c@name
\MT@extra@context
\MT@permutelist

```
2936 \newcommand*\MT@SetExtraKerning[3][]{%
2937     \let\MT@extra@context\@empty
2938     \MT@set@named@keys{MT@kn@c}{#1}%
2939 ⟨debug⟩\MT@dinfo{1}{creating kerning list '\MT@kn@c@name'}%
```

```
2940    \def\MT@permutelist{kn@c}%
2941    \setkeys{MT@cfg}{#2}%
2942    \MT@permute
2943    \MT@gdef@n{MT@kn@c@\MT@kn@c@name}{#3}%
2944    \MT@end@catcodes
2945 }
```

\MT@set@named@keys   We first set the name (if specified), then remove it from the list, and set the
\MT@options   remaining keys.

```
2946 \def\MT@set@named@keys#1#2{%
2947    \def\x##1name=##2,##3\@nil{%
2948      \setkeys{#1}{name=##2}%
2949      \gdef\MT@options{##1##3}%
2950      \MT@rem@from@clist{name=}\MT@options
2951    }%
2952    \x#2,name=,\@nil
2953    \@expandtwoargs\setkeys{#1}\MT@options
2954 }
```

\MT@define@code@key   Define the keys for the configuration lists (which are setting the codes, in pdfTEX
                      speak).

```
2955 \def\MT@define@code@key#1#2{%
2956    \define@key{MT@#2}{#1}[]{%
2957      \@tempcnta=\@ne
2958      \MT@map@clist@n{##1}{%
2959        \KV@@sp@def\MT@val{####1}%
```

Here, too, we allow for something like 'bf*'. It will be expanded immediately.

```
2960        \MT@get@highlevel{#1}%
2961        \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
2962        \advance\@tempcnta \@ne
2963      }%
2964    }%
2965 }
```

\MT@define@code@key@size   \MT@tempsize must be in a \csname, so that it is at least \relax, not undefined.

```
2966 \def\MT@define@code@key@size#1{%
2967    \define@key{MT@#1}{size}[]{%
2968      \MT@map@clist@n{##1}{%
2969        \KV@@sp@def\MT@val{####1}%
2970        \expandafter\MT@get@range\MT@val--\@nil
2971        \ifx\MT@val\relax \else
2972          \MT@exp@cs\MT@xadd{MT@tempsize}%
2973            {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2974        \fi
2975      }%
2976    }%
2977 }
```

\MT@define@code@key@font

```
2978 \def\MT@define@code@key@font#1{%
2979    \define@key{MT@#1}{font}[]{%
2980      \MT@map@clist@n{##1}{%
2981        \KV@@sp@def\MT@val{####1}%
2982        \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/*}}\relax
2983        \expandafter\MT@get@font@and@size\MT@val/////\@nil
2984        \MT@xdef@n{MT@\MT@permutelist @\@tempb\MT@extra@context}%
2985          {\csname MT@\MT@permutelist @name\endcsname}%
2986 ⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempb=\MT@val
2987 ⟨debug⟩                    \ifx\MT@extra@context\@empty\else\MessageBreak
```

```
2988 ⟨debug⟩                        (context: \MT@extra@context)\fi}%
2989     \MT@exp@cs\MT@xaddb
2990       {MT@\MT@permutelist @\@tempb\MT@extra@context @sizes}%
2991       {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
2992     }%
2993   }%
2994 }
```

\MT@get@font@and@size   Translate any asterisks and split off the size.

```
2995 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2996   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
2997 }
```

```
2998 \MT@define@code@key{encoding}{cfg}
2999 \MT@define@code@key{family}  {cfg}
3000 \MT@define@code@key{series}  {cfg}
3001 \MT@define@code@key{shape}   {cfg}
3002 \MT@define@code@key@size      {cfg}
3003 \MT@define@code@key@font      {cfg}
```

\MT@define@opt@key

```
3004 \def\MT@define@opt@key#1#2{%
3005   \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
3006     \MT@xdef@n{MT@#1@c@\MT@curr@set@name @#2}{##1}}}%
3007 }
```

The options in the optional first argument.

```
3008 \MT@map@clist@c\MT@features{%
```

Use file name and line number as the list name if the user didn't bother to invent one.

```
3009   \define@key{MT@#1@c}{name}[]{%
3010     \MT@ifempty{##1}{%
3011       \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
3012     }{%
3013       \MT@edef@n{MT@#1@c@name}{##1}%
3014       \MT@ifdefined@n@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%
3015         \MT@warning{Redefining \@nameuse{MT@abbr@#1} list '\@nameuse{MT@#1@c@name}'}%
3016       }%
3017     }%
3018     \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
3019   }%
3020   \MT@define@opt@key{#1}{load}%
3021   \MT@define@opt@key{#1}{factor}%
3022   \MT@define@opt@key{#1}{preset}%
3023   \MT@define@opt@key{#1}{inputenc}%
```

Only one context is allowed. This might change in the future.

```
3024   \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
3025 }
```

Automatically enable font copying if we find a protrusion or expansion context. After the preamble, check whether font copying is enabled. For older pdfTeX versions, disallow. Also disable for luaTeX.

```
3026 \MT@requires@pdftex7{
3027 ⟨∗lua⟩
3028   \MT@requires@luatex{
3029     \define@key{MT@ex@c}{context}[]{%
3030       \MT@error{Expansion contexts currently don't work with luatex.\MessageBreak
3031         Ignoring 'context' key\on@line}%
```

```
3032              {Use pdftex instead.}%
3033        }
3034     }{
3035  ⟨/lua⟩
3036      \define@key{MT@ex@c}{context}[]{%
3037        \MT@ifempty{#1}\relax{%
3038          \MT@glet\MT@copy@font\MT@copy@font@
3039          \def\MT@extra@context{#1}%
3040        }%
3041      }
3042      \MT@addto@setup{%
3043        \define@key{MT@ex@c}{context}[]{%
3044          \ifx\MT@copy@font\MT@copy@font@
3045            \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3046          \else
3047            \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
3048               Ignoring `context' key\on@line}%
3049              {Either move the settings inside the preamble,\MessageBreak
3050                or load the package with the `copyfonts' option.}%
3051          \fi
3052        }%
3053      }
```

Protrusion contexts *may* also work without copying the font, so we don't issue
an error but only a warning. The problem is that pdfTeX only allows one set of
protrusion factors for a given font within one paragraph (those that are in effect at
the end of the paragraph will be in effect for the whole paragraph). When different
fonts are loaded – like in the example with the footnote markers – we don't need
to copy the fonts.

```
3054      \define@key{MT@pr@c}{context}[]{%
3055        \MT@ifempty{#1}\relax{%
3056          \MT@glet\MT@copy@font\MT@copy@font@
3057          \def\MT@extra@context{#1}%
3058        }%
3059      }
3060      \MT@addto@setup{%
3061        \define@key{MT@pr@c}{context}[]{%
3062          \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3063          \ifx\MT@copy@font\MT@copy@font@\else
3064            \MT@warning@nl{If protrusion contexts don't work as expected,
3065               \MessageBreak load the package with the `copyfonts' option}%
3066          \fi
3067        }%
3068      }
3069  ⟨lua⟩  }
3070    }{
3071      \define@key{MT@ex@c}{context}[]{%
3072        \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
3073             or later. Ignoring `context' key\on@line}%
3074          {Upgrade pdftex.}%
3075      }
3076  }
```

\MT@warn@nodim

```
3077  \def\MT@warn@nodim#1{%
3078    \MT@warning{`\@tempa' is not a dimension.\MessageBreak
3079                 Ignoring it and setting values relative to\MessageBreak #1}%
3080  }
```

Protrusion codes may be relative to character width, or to any dimension.

```
3081 \define@key{MT@pr@c}{unit}[character]{%
3082   \MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}\@empty
3083   \def\@tempa{#1}%
3084   \MT@ifstreq\@tempa{character}\relax{%
```

Test whether it's a dimension, but do not translate it into its final form here, since it may be font-specific.

```
3085     \MT@ifdimen\@tempa
3086       {\MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}\@tempa}%
3087       {\MT@warn@nodim{character widths}}%
3088   }%
3089 }
```

Tracking may only be relative to a dimension.

```
3090 \define@key{MT@tr@c}{unit}[1em]{%
3091   \MT@glet@nc{MT@tr@c@\MT@curr@set@name @unit}\@empty
3092   \def\@tempa{#1}%
3093   \MT@ifdimen\@tempa
3094     {\MT@glet@nc{MT@tr@c@\MT@curr@set@name @unit}\@tempa}%
3095     {\MT@warn@nodim{1em}%
3096      \MT@gdef@n{MT@tr@c@\MT@curr@set@name @unit}{1em}}%
3097 }
```

Spacing and kerning codes may additionally be relative to space dimensions.

```
3098 \MT@map@clist@n{sp,kn}{%
3099   \define@key{MT@#1@c}{unit}[space]{%
3100     \MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\@empty
3101     \def\@tempa{##1}%
3102     \MT@ifstreq\@tempa{character}\relax{%
3103       \MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\m@ne
3104       \MT@ifstreq\@tempa{space}\relax{%
3105         \MT@ifdimen\@tempa
3106           {\MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\@tempa}%
3107           {\MT@warn@nodim{width of space}}%
3108       }%
3109     }%
3110   }%
3111 }
```

The first argument to \SetExpansion accepts some more options.

```
3112 \MT@map@clist@n{stretch,shrink,step}{%
3113   \define@key{MT@ex@c}{#1}[]{%
3114     \MT@ifempty{##1}\relax{%
3115       \MT@ifint{##1}{%
```

A space terminates the number.

```
3116         \MT@gdef@n{MT@ex@c@\MT@curr@set@name @#1}{##1 }%
3117       }{%
3118         \MT@warning{%
3119           Value '##1' for option '#1' is not a number.\MessageBreak
3120           Ignoring it}%
3121       }%
3122     }%
3123   }%
3124 }
3125 \define@key{MT@ex@c}{auto}[true]{%
3126   \def\@tempa{#1}%
3127   \csname if\@tempa\endcsname
```

Don't use `autoexpand` for pdfTEX version older than 1.20.

```
3128     \MT@requires@pdftex4{%
3129       \MT@gdef@n{MT@ex@c@\MT@curr@set@name @auto}{autoexpand}%
3130     }{%
3131       \MT@warning{pdftex too old for automatic font expansion}%
3132     }
3133   \else
3134     \MT@requires@pdftex4{%
3135       \MT@glet@nc{MT@ex@c@\MT@curr@set@name @auto}\@empty
3136     }\relax
3137   \fi
3138 }
```

Tracking: Interword spacing and outer kerning. The variant with space in case `\SetTracking` is called inside an argument (e. g., to `\IfFileExists`).

```
3139 \MT@define@opt@key{tr}{spacing}
3140 \MT@define@opt@key{tr}{outerspacing}
3141 \MT@define@opt@key{tr}{outerkerning}
```

Which ligatures should be disabled?

```
3142 \define@key{MT@tr@c}{noligatures}[]%
3143   {\MT@xdef@n{MT@tr@c@\MT@curr@set@name @noligatures}{#1}}
3144 \define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
3145 \define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
3146 \define@key{MT@tr@c}{no ligatures}[]{\setkeys{MT@tr@c}{noligatures={#1}}}
```

14.3.6 Character inheritance

`\DeclareCharacterInheritance` This macro may be used in the configuration files to declare characters that should inherit protrusion resp. expansion values from other characters. Thus, there is no need to define all accented characters (e. g., `\'a, \'a, \^a, \~a, \"a, \r{a}, \k{a}, \u{a}`), which will make the configuration files look much nicer and easier to maintain. If a single character of an inheritance list should have a different value, one can simply override it.

`\MT@inh@feat`   The optional argument may be used to restrict the list to some features,
`\MT@extra@inputenc`   and to specify an input encoding.

```
3147 \renewcommand*\DeclareCharacterInheritance[1][]{%
3148   \let\MT@extra@context\@empty
3149   \let\MT@extra@inputenc\@undefined
3150   \let\MT@inh@feat\@empty
3151   \setkeys{MT@inh@}{#1}%
3152   \MT@begin@catcodes
3153   \MT@set@inh@list
3154 }
```

`\MT@set@inh@list`   Safe category codes.

```
3155 \def\MT@set@inh@list#1#2{%
3156   \MT@ifempty\MT@inh@feat{%
3157     \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{#1}{#2}}}%
3158   }{%
3159     \MT@map@clist@c\MT@inh@feat{{%
3160       \KV@@sp@def\@tempa{##1}%
3161       \MT@ifempty\@tempa\relax{%
3162         \MT@exp@one@n\MT@declare@char@inh
3163           {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
3164       }%
3165     }}%
```

```
3166    }%
3167    \MT@end@catcodes
3168 }
```

The keys for the optional argument.

```
3169 \MT@map@clist@c\MT@features@long{%
3170    \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}}}
3171 \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}
```

\MT@declare@char@inh   The lists cannot be given a name by the user.

```
3172 \def\MT@declare@char@inh#1#2#3{%
3173    \MT@edef@n{MT@#1@inh@name}%
3174      {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3175    \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
3176    \MT@ifdefined@c@T\MT@extra@inputenc{%
3177      \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
3178 ⟨debug⟩\MT@dinfo{1}{creating inheritance list '\@nameuse{MT@#1@inh@name}'}%
3179    \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
3180    \def\MT@permutelist{#1@inh}%
3181    \setkeys{MT@inh}{#2}%
3182    \MT@permute
3183 }
```

Parse the second argument. \DeclareCharacterInheritance may also be set up
for various combinations.

```
3184 \define@key{MT@inh}{encoding}[]{%
3185    \def\MT@val{#1}%
3186    \expandafter\MT@encoding@check\MT@val,\@nil
3187    \MT@get@highlevel{encoding}%
3188    \MT@edef@n{MT@tempencoding1}{\MT@val}%
3189 }
```

\MT@encoding@check   But we only allow *one* encoding.

```
3190 \def\MT@encoding@check#1,#2\@nil{%
3191    \MT@ifempty{#2}\relax{%
3192      \edef\MT@val{#1}%
3193      \MT@warning{You may only specify one encoding for character\MessageBreak
3194                  inheritance lists. Ignoring encoding(s) #2}%
3195    }%
3196 }
```

For the rest, we can reuse the key setup from the configuration lists (\Set...).

```
3197 \MT@define@code@key{family}{inh}
3198 \MT@define@code@key{series}{inh}
3199 \MT@define@code@key{shape} {inh}
3200 \MT@define@code@key@size    {inh}
3201 \MT@define@code@key@font    {inh}
```

\MT@inh@do   Now parse the third argument, the inheritance lists. We define the commands
\MT@inh@⟨name⟩@⟨slot⟩@, containing the inheriting characters. They will also be
translated to slot numbers here, to save some time. The following will be exe-
cuted only once, namely the first time this inheritance list is encountered (in
\MT@set@⟨feature⟩@codes).

```
3202 \def\MT@inh@do#1,{%
3203    \ifx\relax#1\@empty \else
3204      \MT@inh@split #1==\relax
3205      \expandafter\MT@inh@do
3206    \fi
3207 }
```

\MT@inh@split   Only gather the inheriting characters here. Their codes will actually be set in
                       \MT@set@⟨feature⟩@codes.

```
3208 \def\MT@inh@split#1=#2=#3\relax{%
3209   \def\@tempa{#1}%
3210   \ifx\@tempa\@empty \else
3211     \MT@get@slot
3212     \ifnum\MT@char > \m@ne
3213       \let\MT@val\MT@char
3214       \MT@map@clist@n{#2}{%
3215         \def\@tempa{##1}%
3216         \ifx\@tempa\@empty \else
3217           \MT@get@slot
3218           \ifnum\MT@char > \m@ne
3219             \MT@exp@cs\MT@xadd{MT@inh@\MT@listname @\MT@val @}{{\MT@char}}%
3220           \fi
3221         \fi
3222       }%
3223 ⟨debug⟩\MT@dinfo@nl{2}{children of #1 (\MT@val):
3224 ⟨debug⟩                    \@nameuse{MT@inh@\MT@listname @\MT@val @}}%
3225     \fi
3226   \fi
3227 }
```

14.3.7 Permutation

\MT@permute   Calling \MT@permute will define commands for all permutations of the specified
\MT@permute@   font attributes of the form \MT@⟨list type⟩@/⟨encoding⟩/⟨family⟩/⟨series⟩/⟨shape⟩/
\MT@permute@@   ⟨/∗⟩ to be the expansion of \MT@⟨list type⟩@name, i.e., the name of the currently
\MT@permute@@@   defined list. Size ranges are held in a separate macro called \MT@⟨list type⟩@/⟨font
\MT@permute@@@@   axes⟩@sizes, which in turn contains the respective ⟨list name⟩s attached to the
                       ranges.

```
3228 \def\MT@permute{%
3229   \let\MT@cnt@encoding\@ne
3230   \MT@permute@
```

Undefine commands for the next round.

```
3231   \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
3232   \MT@glet\MT@tempsize\@undefined
3233 }
3234 \def\MT@permute@{%
3235   \let\MT@cnt@family\@ne
3236   \MT@permute@@
3237   \MT@increment\MT@cnt@encoding
3238   \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
3239     \MT@permute@
3240 }
3241 \def\MT@permute@@{%
3242   \let\MT@cnt@series\@ne
3243   \MT@permute@@@
3244   \MT@increment\MT@cnt@family
3245   \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
3246     \MT@permute@@
3247 }
3248 \def\MT@permute@@@{%
3249   \let\MT@cnt@shape\@ne
3250   \MT@permute@@@@
3251   \MT@increment\MT@cnt@series
```

```
3252     \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
3253       \MT@permute@@@
3254 }
3255 \def\MT@permute@@@@{%
3256    \MT@permute@@@@@
3257    \MT@increment\MT@cnt@shape
3258    \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
3259       \MT@permute@@@@
3260 }
```

\MT@permute@@@@@   In order to save some memory, we can ignore unused encodings (inside the document).

```
3261 \def\MT@permute@@@@@{%
3262   \MT@permute@define{encoding}%
3263   \ifMT@document
3264     \ifx\MT@tempencoding\@empty \else
3265       \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
3266          {\expandafter\expandafter\expandafter\@gobble}%
3267     \fi
3268   \fi
3269   \MT@permute@@@@@@
3270 }
```

\MT@permute@@@@@@

```
3271 \def\MT@permute@@@@@@{%
3272   \MT@permute@define{family}%
3273   \MT@permute@define{series}%
3274   \MT@permute@define{shape}%
3275   \edef\@tempa{\MT@tempencoding
3276              /\MT@tempfamily
3277              /\MT@tempseries
3278              /\MT@tempshape
3279              /\MT@ifdefined@c@T\MT@tempsize *}%
```

Some sanity checks: an encoding must be specified (unless nothing else is).

```
3280    \MT@ifstreq\@tempa{////}\relax{%
3281      \ifx\MT@tempencoding\@empty
3282        \MT@warning{%
3283          You have to specify an encoding for\MessageBreak
3284          \@nameuse{MT@abbr@\MT@permutelist} list
3285          '\@nameuse{MT@\MT@permutelist @name}'.\MessageBreak
3286          Ignoring it}%
3287      \else
3288        \MT@ifdefined@c@TF\MT@tempsize{%
```

Add the list of ranges to the beginning of the current combination, after checking for conflicts.

```
3289          \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3290            \MT@map@tlist@c\MT@tempsize\MT@check@rlist
3291          }%
3292          \MT@exp@cs\MT@xaddb
3293            {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
3294            \MT@tempsize
3295 ⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempa,\MessageBreak
3296 ⟨debug⟩          sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
3297 ⟨debug⟩                    @sizes\endcsname}%
3298        }{%
```

Only one list can apply to a given combination.

```
3299          \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context}{%
```

```
3300            \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
3301                '\@nameuse{MT@\MT@permutelist @name}' will override list\MessageBreak
3302                '\@nameuse{MT@\MT@permutelist @\@tempa\MT@extra@context}'
3303                for font '\@tempa'}%
3304            }%
3305 ⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempa
3306 ⟨debug⟩                    \ifx\MT@extra@context\@empty\else\MessageBreak
3307 ⟨debug⟩                       (context: \MT@extra@context)\fi}%
3308        }%
3309        \MT@xdef@n{MT@\MT@permutelist @\@tempa\MT@extra@context}%
3310            {\csname MT@\MT@permutelist @name\endcsname}%
3311      \fi
3312    }%
3313 }
```

\MT@permute@define   Define the commands.

```
3314 \def\MT@permute@define#1{%
3315   \@tempcnta=\csname MT@cnt@#1\endcsname\relax
3316   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3317     {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3318     {\MT@let@nc{MT@temp#1}\@empty}%
3319 }
```

\MT@permute@reset   Reset the commands.

```
3320 \def\MT@permute@reset#1{%
3321   \@tempcnta=\@ne
3322   \MT@loop
3323     \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3324     \advance\@tempcnta\@ne
3325     \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3326        \iftrue
3327        \iffalse
3328    \MT@repeat
3329 }
```

\MT@check@rlist   For every new range item in \MT@tempsize, check whether it overlaps with ranges in the existing list.

```
3330 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}
```

\MT@check@rlist@   Define the current new range and . . .

```
3331 \def\MT@check@rlist@#1#2#3{%
3332   \def\@tempb{#1}%
3333   \def\@tempc{#2}%
3334   \MT@if@false
3335   \MT@exp@cs\MT@map@tlist@c
3336     {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
3337     \MT@check@range
3338 }
```

\MT@check@range   . . . recurse through the list of existing ranges.

```
3339 \def\MT@check@range#1{\expandafter\MT@check@range@ #1}
```

\MT@check@range@   \@tempb and \@tempc are lower resp. upper bound of the new range, ⟨#2⟩ and ⟨#3⟩ those of the existing range.

```
3340 \def\MT@check@range@#1#2#3{%
3341   \MT@ifdim{#2}=\m@ne{%
3342     \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
3343        \MT@ifdim\@tempb={#1}\MT@if@true\relax
3344      }{%
```

- Item in list is a simple size, new item is a range.

```
3345        \MT@ifdim\@tempb>{#1}\relax{%
3346          \MT@ifdim\@tempc>{#1}{%
3347            \MT@if@true
3348            \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3349          }\relax
3350        }%
3351      }%
3352    }{%
3353      \MT@ifdim\@tempc=\m@ne{%
```

- Item in list is a range, new item is a simple size.

```
3354        \MT@ifdim\@tempb<{#2}{%
3355          \MT@ifdim\@tempb<{#1}\relax\MT@if@true
3356        }\relax
3357      }{%
```

- Both items are ranges.

```
3358        \MT@ifdim\@tempb<{#2}{%
3359          \MT@ifdim\@tempc>{#1}{%
3360            \MT@if@true
3361            \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
3362          }\relax
3363        }\relax
3364      }%
3365    }%
3366    \ifMT@if@
3367      \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
3368        '\@nameuse{MT@\MT@permutelist @name}' will override\MessageBreak
3369        list '#3' for font \@tempa,\MessageBreak size \@tempb}%
```

If we've already found a conflict with this item, we can skip the rest of the list.

```
3370      \expandafter\MT@tlist@break
3371    \fi
3372 }
```

14.4 Package options

14.4.1 Declaring the options

\ifMT@opt@expansion   Keep track of whether the user explicitly set these options.
\ifMT@opt@auto
\ifMT@opt@DVI
```
3373 \newif\ifMT@opt@expansion
3374 \newif\ifMT@opt@auto
3375 \newif\ifMT@opt@DVI
```

\MT@optwarn@admissible   Some warnings.
```
3376 \def\MT@optwarn@admissible#1#2{%
3377   \MT@warning@nl{'#1' is not an admissible value for option\MessageBreak
3378                  '#2'. Assuming 'false'}%
3379 }
```

\MT@optwarn@nan

```
3380 ⟨/package⟩
```

```
3381 ⟨plain⟩\MT@requires@latex1{
3382 \def\MT@optwarn@nan#1#2{%
3383   \MT@warning@nl{Value '#1' for option '#2' is not a\MessageBreak number.
3384                   Using default value of \number\@nameuse{MT@#2@default}}%
3385 }
3386 ⟨plain⟩}\relax
3387 ⟨∗package⟩
```

\MT@opt@def@set

```
3388 \def\MT@opt@def@set#1{%
3389   \MT@ifdefined@n@TF{MT@\@tempb @set@@\MT@val}{%
3390     \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
3391   }{%
3392     \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
3393     \MT@warning@nl{The #1 set '\MT@val' is undeclared.\MessageBreak
3394                   Using set '\@nameuse{MT@\@tempb @setname}' instead}%
3395   }%
3396 }
```

expansion and protrusion may be true, false, compatibility, nocompatibility and/or a ⟨set name⟩.

```
3397 \MT@map@clist@n{protrusion,expansion}{%
3398   \define@key{MT}{#1}[true]{%
3399     \csname MT@opt@#1true\endcsname
3400     \MT@map@clist@n{##1}{%
3401       \KV@@sp@def\MT@val{####1}%
3402       \MT@ifempty\MT@val\relax{%
3403         \csname MT@#1true\endcsname
3404         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3405         \MT@ifstreq\MT@val{true}\relax
3406         {%
3407           \MT@ifstreq\MT@val{false}{%
3408             \csname MT@#1false\endcsname
3409           }{%
3410             \MT@ifstreq\MT@val{compatibility}{%
3411               \MT@let@nc{MT@\@tempb @level}\@ne
3412             }{%
3413               \MT@ifstreq\MT@val{nocompatibility}{%
3414                 \MT@let@nc{MT@\@tempb @level}\tw@
3415               }{%
```

If everything failed, it should be a set name.

```
3416                 \MT@opt@def@set{#1}%
3417               }%
3418             }%
3419           }%
3420         }%
3421       }%
3422     }%
3423   }%
3424 }
```

activate is a shortcut for protrusion and expansion.

```
3425 \define@key{MT}{activate}[true]{%
3426   \setkeys{MT}{protrusion={#1}}%
3427   \setkeys{MT}{expansion={#1}}%
3428 }
```

spacing, kerning and tracking do not have a compatibility level.

```
3429 \MT@map@clist@n{spacing,kerning,tracking}{%
```

```
3430    \define@key{MT}{#1}[true]{%
3431      \MT@map@clist@n{##1}{%
3432        \KV@@sp@def\MT@val{####1}%
3433        \MT@ifempty\MT@val\relax{%
3434          \csname MT@#1true\endcsname
3435          \MT@ifstreq\MT@val{true}\relax
3436          {%
3437            \MT@ifstreq\MT@val{false}{%
3438              \csname MT@#1false\endcsname
3439            }{%
3440              \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3441              \MT@opt@def@set{#1}%
3442            }%
3443          }%
3444        }%
3445      }%
3446    }%
3447  }
```

\MT@def@bool@opt   The true/false options: draft, final (may be inherited from the class options), auto, selected, babel, DVIoutput, defersetup, copyfonts.

```
3448  \def\MT@def@bool@opt#1#2{%
3449    \define@key{MT}{#1}[true]{%
3450      \def\@tempa{##1}%
3451      \MT@ifstreq\@tempa{true}\relax{%
3452        \MT@ifstreq\@tempa{false}\relax{%
3453          \MT@optwarn@admissible{##1}{#1}%
3454          \def\@tempa{false}%
3455        }%
3456      }%
3457      #2%
3458    }%
3459  }
```

Boolean options that only set the switch.

```
3460  \MT@map@clist@n{draft,selected,babel}{%
3461    \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
3462  \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotrue}
```

The DVIoutput option will change \pdfoutput immediately to minimise the risk of confusing other packages.

```
3463  \MT@def@bool@opt{DVIoutput}{%
3464    \csname if\@tempa\endcsname
3465      \ifnum\pdfoutput>\z@ \MT@opt@DVItrue \fi
3466      \pdfoutput\z@
3467    \else
3468      \ifnum\pdfoutput<\@ne \MT@opt@DVItrue \fi
3469      \pdfoutput\@ne
3470    \fi
3471  }
```

Setting the defersetup option to false will restore the old behaviour, where the setup took place at the time when the package was loaded. This is undocumented, since I would like to learn about the cases where this is necessary.

The only problem with the new deferred setup I can think of is when a box is being constructed inside the preamble and this box contains a font that is not loaded before the box is being used.

```
3472  \MT@def@bool@opt{defersetup}{%
```

```
3473    \csname if\@tempa\endcsname \else
3474      \AtEndOfPackage{%
3475        \MT@setup@
3476        \let\MT@setup@\@empty
3477        \let\MT@addto@setup\@firstofone
3478      }%
3479    \fi
3480 }
```

`copyfonts` will copy all fonts before setting them up. This allows protrusion and expansion with different parameters. This options is also *undocumented* in the hope that we can always find out automatically whether it's required.

```
3481 \MT@requires@pdftex7{
3482 ⟨*lua⟩
3483    \MT@requires@luatex{
3484      \MT@def@bool@opt{copyfonts}{%
3485        \csname if\@tempa\endcsname
3486          \MT@error{The 'copyfonts' option doesn't work with luatex}
3487                  {Use pdftex instead.}%
3488        \fi
3489      }
3490    }{
3491 ⟨/lua⟩
3492      \MT@def@bool@opt{copyfonts}{%
3493        \csname if\@tempa\endcsname
3494          \MT@glet\MT@copy@font\MT@copy@font@
3495        \else
3496          \MT@glet\MT@copy@font\relax
3497        \fi
3498      }
3499 ⟨lua⟩  }
3500 }{
3501    \MT@def@bool@opt{copyfonts}{%
3502      \csname if\@tempa\endcsname
3503        \MT@error{The pdftex version you are using is too old\MessageBreak
3504          to use the 'copyfonts' option}{Upgrade pdftex.}%
3505      \fi
3506    }
3507 }
```

`final` is the opposite to `draft`.

```
3508 \MT@def@bool@opt{final}{%
3509   \csname if\@tempa\endcsname
3510     \MT@draftfalse
3511   \else
3512     \MT@drafttrue
3513   \fi
3514 }
```

For `verbose` output, we redefine `\MT@vinfo`.

```
3515 \define@key{MT}{verbose}[true]{%
3516   \let\MT@vinfo\MT@info@nl
3517   \def\@tempa{#1}%
3518   \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```
3519     \MT@ifstreq\@tempa{errors}{%
3520       \let\MT@warning    \MT@warn@err
3521       \let\MT@warning@nl\MT@warn@err
3522     }{%
```

```
3523        \let\MT@vinfo\@gobble
```

Cast warnings to the winds.

```
3524        \MT@ifstreq\@tempa{silent}{%
3525          \let\MT@warning    \MT@info
3526          \let\MT@warning@nl\MT@info@nl
3527        }{%
3528          \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
3529        }%
3530      }%
3531    }%
3532 }
```

Options with numerical keys: `factor`, `stretch`, `shrink`, `step`, `letterspace`.

```
3533 ⟨/package⟩
3534 ⟨plain⟩\MT@requires@latex1{
3535 \MT@map@clist@n{%
3536 ⟨package⟩      stretch,shrink,step,%
3537      letterspace}{%
3538   \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
3539      \def\@tempa{##1 }%
```

No nonsense in `\MT@factor` et al.? A space terminates the number.

```
3540      \MT@ifint\@tempa
3541        {\MT@edef@n{MT@#1}{\@tempa}}%
3542        {\MT@optwarn@nan{##1}{#1}}%
3543    }%
3544 }
3545 ⟨plain⟩}\relax
3546 ⟨*package⟩
```

`factor` will define the protrusion factor only.

```
3547 \define@key{MT}{factor}[\MT@factor@default]{%
3548   \def\@tempa{#1 }%
3549   \MT@ifint\@tempa
3550     {\edef\MT@pr@factor{\@tempa}}
3551     {\MT@optwarn@nan{#1}{factor}}%
3552 }
```

Unit for protrusion codes.

```
3553 \define@key{MT}{unit}[character]{%
3554   \def\@tempa{#1}%
3555   \MT@ifstreq\@tempa{character}\relax{%
3556     \MT@ifdimen\@tempa
3557       {\let\MT@pr@unit\@tempa}%
3558       {\MT@warning@nl{'\@tempa' is not a dimension.\MessageBreak
3559              Ignoring it and setting values relative to\MessageBreak
3560              character widths}}%
3561    }%
3562 }
```

14.4.2   Reading the configuration file

The package should just work if called without any options. Therefore, expansion will be switched off by default if output is DVI, since it isn't likely that expanded fonts are available. (This grows more important as modern TeX systems have switched to the pdfTeX engine even for DVI output, so that the user might not even be aware of the fact that she's running pdfTeX.)

```
3563 \MT@protrusiontrue
3564 \ifnum\pdfoutput<\@ne \else
```

Also, we only enable expansion by default if pdfTeX can expand the fonts automatically.

```
3565   \MT@requires@pdftex4{
3566     \MT@expansiontrue
3567     \MT@autotrue
3568   }\relax
3569 \fi
```

The main configuration file will be loaded before processing the package options. However, the config option must of course be evaluated beforehand. We also have to define a no-op for the regular option processing later.

\MT@config@file
\MT@get@config

```
3570 \define@key{MT}{config}[]{\relax}
3571 \def\MT@get@config#1config=#2,#3\@nil{%
3572   \MT@ifempty{#2}%
3573     {\def\MT@config@file{\MT@MT.cfg}}%
3574     {\def\MT@config@file{#2.cfg}}%
3575 }
3576 \expandafter\expandafter\expandafter\MT@get@config
3577   \csname opt@\@currname.\@currext\endcsname,config=,\@nil
```

Load the file.

```
3578 \IfFileExists{\MT@config@file}{%
3579   \MT@info@nl{Loading configuration file \MT@config@file}%
3580   \MT@begin@catcodes
3581     \let\MT@begin@catcodes\relax
3582     \let\MT@end@catcodes\relax
3583     \let\MT@curr@file\MT@config@file
3584     \input{\MT@config@file}%
3585   \endgroup
3586 }{\MT@warning@nl{%
3587     Could not find configuration file '\MT@config@file'!\MessageBreak
3588     This will almost certainly cause undesired results.\MessageBreak
3589     Please fix your installation}%
3590 }
```

\MT@check@active@set

We have to make sure that font sets are active. If the user didn't activate any, we use those sets declared by \DeclareMicrotypeSetDefault (this is done at the end of the preamble).

```
3591 \def\MT@check@active@set#1{%
3592   \MT@ifdefined@n@TF{MT@#1@setname}{%
3593     \MT@info@nl{Using \@nameuse{MT@abbr@#1} set '\@nameuse{MT@#1@setname}'}%
3594   }{%
3595     \MT@ifdefined@n@TF{MT@default@#1@set}{%
3596       \MT@glet@nn{MT@#1@setname}{MT@default@#1@set}%
3597       \MT@info@nl{Using default \@nameuse{MT@abbr@#1} set '\@nameuse{MT@#1@setname}'}%
3598     }{%
```

If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set '@', and issue a warning.

```
3599       \MT@gdef@n{MT@#1@setname}{@}%
3600       \MT@warning@nl{No \@nameuse{MT@abbr@#1} set chosen, no default set declared.
3601                     \MessageBreak Using empty set}%
3602     }%
3603   }%
3604 }
```

14.4.3 Hook for other packages

\Microtype@Hook

This hook may be used by font package authors, e.g., to declare alias fonts. If it is defined, it will be executed here, i.e., after the main configuration file has been loaded, and before the package options are evaluated.

This hook was needed in versions prior to 1.9a to overcome the situation that (1) the microtype package should be loaded after all font defaults have been set up (hence, using \@ifpackageloaded in the font package was not viable), and (2) checking \AtBeginDocument could be too late, since fonts might already have been loaded, and consequently set up, in the preamble. With the new deferred setup, one could live without this command, however, it remains here since it's simpler than testing whether the package was loaded both in the preamble as well as at the beginning of the document (which is what one would have to do).

Package authors should check whether the command is already defined so that existing definitions by other packages aren't overwritten. Example:

```
\def\MinionPro@MT@Hook{\DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}}
\@ifpackageloaded{microtype}
  \MinionPro@MT@Hook
  {\@ifundefined{Microtype@Hook}
    {\let\Microtype@Hook\MinionPro@MT@Hook}
    {\g@addto@macro\Microtype@Hook{\MinionPro@MT@Hook}}}
```

\MicroType@Hook with a capital T (which only existed in version 1.7) is provided for compatibility reasons. At some point in the future, it will no longer be available, hence it should not be used.

```
3605 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
3606   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3607       Use \string\Microtype@Hook\space instead}\MicroType@Hook}
3608 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook
```

14.4.4 Changing options later

\microtypesetup
\MT@define@optionX

Inside the preamble, \microtypesetup accepts the same options as the package (unless defersetup=false). In the document body, it accepts the options: protrusion, expansion, activate, tracking, spacing and kerning. Specifying font sets is not allowed.

```
3609 \def\microtypesetup{\setkeys{MT}}
3610 \MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
3611 \def\MT@define@optionX#1#2{%
3612   \define@key{MTX}{#1}[true]{%
3613     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3614     \MT@map@clist@n{##1}{%
3615       \KV@@sp@def\MT@val{####1}%
3616       \MT@ifempty\MT@val\relax{%
3617         \@tempcnta=\m@ne
3618         \MT@ifstreq\MT@val{true}{%
```

Enabling micro-typography in the middle of the document is not allowed if it has been disabled in the package options since fonts might already have been loaded and hence wouldn't be set up.

```
3619         \MT@checksetup{#1}{%
3620           \@tempcnta=\csname MT@\@tempb @level\endcsname
3621           \MT@vinfo{Enabling #1
```

```
3622                    (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
3623                }%
3624             }{%
3625               \MT@ifstreq\MT@val{false}{%
3626                 \@tempcnta=\z@
3627                 \MT@vinfo{Disabling #1\on@line}%
3628               }{%
3629                 \MT@ifstreq\MT@val{compatibility}{%
3630                   \MT@checksetup{#1}{%
3631                     \@tempcnta=\@ne
3632                     \MT@let@nc{MT@\@tempb @level}\@ne
3633                     \MT@vinfo{Setting #1 to level 1\on@line}%
3634                   }%
3635                 }{%
3636                   \MT@ifstreq\MT@val{nocompatibility}{%
3637                     \MT@checksetup{#1}{%
3638                       \@tempcnta=\tw@
3639                       \MT@let@nc{MT@\@tempb @level}\tw@
3640                       \MT@vinfo{Setting #1 to level 2\on@line}%
3641                     }%
3642                   }{\MT@error{Value '\MT@val' for key '#1' not recognised}
3643                              {Use any of 'true', 'false', 'compatibility' or
3644                               'nocompatibility'.}%
3645                   }%
3646                 }%
3647               }%
3648             }%
3649             \ifnum\@tempcnta>\m@ne
3650               #2\@tempcnta\relax
3651             \fi
3652           }%
3653         }%
3654     }%
3655 }
```

\MT@checksetup   Test whether the feature wasn't disabled in the package options.

```
3656 \def\MT@checksetup#1{%
3657   \csname ifMT@#1\endcsname
3658     \expandafter\@firstofone
3659   \else
3660     \MT@error{You cannot enable #1 if it was disabled\MessageBreak
3661               in the package options}{Load microtype with #1 enabled.}%
3662     \expandafter\@gobble
3663   \fi
3664 }
```

```
3665 \MT@define@optionX{protrusion}\pdfprotrudechars
3666 \MT@define@optionX{expansion}\pdfadjustspacing
```

\MT@define@optionX@   The same for tracking, spacing and kerning, which do not have a compatibility
level.

```
3667 \MT@requires@pdftex6{
3668 ⟨lua⟩   \MT@requires@luatex\@firstofone{
3669   \def\MT@define@optionX@#1#2{%
3670     \define@key{MTX}{#1}[true]{%
3671       \MT@map@clist@n{##1}{%
3672         \KV@@sp@def\MT@val{####1}%
3673         \MT@ifempty\MT@val\relax{%
3674           \@tempcnta=\m@ne
3675           \MT@ifstreq\MT@val{true}{%
```

```
3676              \MT@checksetup{#1}{%
3677                 \@tempcnta=\@ne
3678                 \MT@vinfo{Enabling #1\on@line}%
3679              }%
3680           }{%
3681              \MT@ifstreq\MT@val{false}{%
3682                 \@tempcnta=\z@
3683                 \MT@vinfo{Disabling #1\on@line}%
3684              }{\MT@error{Value '\MT@val' for key '#1' not recognised}
3685                       {Use either 'true' or 'false'}%
3686              }%
3687           }%
3688           \ifnum\@tempcnta>\m@ne
3689              #2\relax
3690           \fi
3691        }%
3692      }%
3693    }%
3694  }
```

We cannot simply let \MT@tracking relax, since this may select the already letterspaced font instance.

```
3695    \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
3696                                 \else \let\MT@tracking\MT@tracking@ \fi}
3697    \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
3698    \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
3699                                 \pdfappendkern \@tempcnta}
3700    \@gobble
3701 ⟨lua⟩  }
3702 }\@firstofone
```

Disable for older pdfTEX versions and for luaTEX.

```
3703 {\define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
3704  \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
3705  \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
3706 }
3707 \define@key{MTX}{activate}[true]{%
3708   \setkeys{MTX}{protrusion={#1}}%
3709   \setkeys{MTX}{expansion={#1}}%
3710 }
```

\MT@saved@setupfont   Disable everything – may be used as a work-around in case setting up fonts doesn't work in certain environments. *(Undocumented.)*

```
3711 \let\MT@saved@setupfont\MT@setupfont
```

```
3712 \define@key{MTX}{disable}[]{%
3713   \MT@info{Inactivate '\MT@MT' package}%
3714   \let\MT@setupfont\relax
3715 }
3716 \define@key{MTX}{enable}[]{%
3717   \MT@info{Reactivate '\MT@MT' package}%
3718   \let\MT@setupfont\MT@saved@setupfont
3719 }
3720 ⟨/package⟩
```

14.4.5 Processing the options

\MT@ProcessOptionsWithKV   Parse options.

```
3721 ⟨plain⟩\MT@requires@latex1{
```

```
3722 \def\MT@ProcessOptionsWithKV#1{%
3723   \let\@tempc\relax
3724   \let\MT@temp\@empty
3725 ⟨plain⟩   \MT@requires@latex2{
3726      \MT@map@clist@c\@classoptionslist{%
3727        \def\CurrentOption{##1}%
3728        \MT@ifdefined@n@T{KV@#1@\expandafter\MT@getkey\CurrentOption=\@nil}{%
3729          \edef\MT@temp{\MT@temp,\CurrentOption,}%
3730          \@expandtwoargs\@removeelement\CurrentOption
3731            \@unusedoptionlist\@unusedoptionlist
3732        }%
3733      }%
3734      \edef\MT@temp{\noexpand\setkeys{#1}%
3735                        {\MT@temp\@ptionlist{\@currname.\@currext}}}%
```

eplain can handle package options.

```
3736 ⟨*plain⟩
3737   }{\edef\MT@temp{\noexpand\setkeys{#1}%
3738                        {\csname usepkg@options@\usepkg@pkg\endcsname}}}
3739 ⟨/plain⟩
3740   \MT@temp
3741   \MT@clear@options
3742 }
```

\MT@getkey   For key=val in class options.

```
3743 \def\MT@getkey#1=#2\@nil{#1}
```

```
3744 \MT@ProcessOptionsWithKV{MT}
3745 ⟨plain⟩}\relax
3746 ⟨*package⟩
```

Now we can take the appropriate actions. We also tell the log file which options the user has chosen (in case it's interested).

```
3747 \MT@addto@setup{%
3748 \ifMT@draft
```

We disable most of what we've just defined in the 3748 lines above if we are running in draft mode.

```
3749   \MT@warning@nl{'draft' option active.\MessageBreak
3750                    Disabling all micro-typographic extensions.\MessageBreak
3751                    This might lead to different line and page breaks}%
3752   \let\MT@setupfont\relax
3753   \renewcommand*\LoadMicrotypeFile[1]{}%
3754   \renewcommand*\microtypesetup[1]{}%
3755   \renewcommand*\microtypecontext[1]{}%
3756   \renewcommand*\lsstyle{}%
3757 \else
```

For DVI output, the user must have explicitly passed the expansion option to the package.

```
3758   \ifnum\pdfoutput<\@ne
3759     \ifMT@opt@expansion \else
3760       \MT@expansionfalse
3761     \fi
3762   \fi
```

pdfTeX can create DVI output, too. However, both the DVI viewer and dvips need to find actual fonts. Therefore, expansion will only work if the fonts for different degrees of expansion are readily available.

Some packages depend on the value of \pdfoutput and will get confused if it is changed after they have been loaded. These packages are, among others: color, graphics, hyperref, crop, contour, pstricks and, as a matter of course, ifpdf. Instead of testing for each package (that's not our job), we only say that it was microtype that changed it. This must be sufficient!

```
3763    \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3764            \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
```

Working on font copies?

```
3765    \ifx\MT@copy@font\relax\else \MT@info@nl{Using font copies for contexts}\fi
```

Fix the font sets.

```
3766    \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
```

Protrusion.

```
3767    \ifMT@protrusion
3768      \edef\MT@active@features{\MT@active@features,pr}%
3769      \pdfprotrudechars\MT@pr@level
3770      \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
3771        \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3772          factor: \number\MT@pr@factor\fi
3773        \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}%
3774      \MT@check@active@set{pr}%
3775    \else
3776      \let\MT@protrusion\relax
3777      \MT@info@nl{No character protrusion}%
3778    \fi
```

Expansion.

```
3779    \ifMT@expansion
```

Set up the values for font expansion: if stretch has not been specified, we take the default value of 20.

```
3780      \ifnum\MT@stretch=\m@ne
3781        \let\MT@stretch\MT@stretch@default
3782      \fi
```

If shrink has not been specified, it will inherit the value from stretch.

```
3783      \ifnum\MT@shrink=\m@ne
3784        \let\MT@shrink\MT@stretch
3785      \fi
```

If step has not been specified, we will just set it to 1 for recent pdfTeX versions. My tests did not show much difference neither in compilation time (within the margin of error) nor in file size (less than 1% difference for microtype.pdf with step=1 compared to step=5). With older versions, we set it to min(stretch,shrink)/5, rounded off, minimum value 1.

```
3786      \MT@requires@pdftex6{\def\MT@step{1 }}{%
3787        \ifnum\MT@step=\m@ne
3788          \ifnum\MT@stretch>\MT@shrink
3789            \ifnum\MT@shrink=\z@
3790              \@tempcnta=\MT@stretch
3791            \else
3792              \@tempcnta=\MT@shrink
3793            \fi
3794          \else
3795            \ifnum\MT@stretch=\z@
3796              \@tempcnta=\MT@shrink
```

```
3797            \else
3798              \@tempcnta=\MT@stretch
3799            \fi
3800          \fi
3801          \divide\@tempcnta 5\relax
3802        \else
3803          \@tempcnta=\MT@step
3804          \ifnum\@tempcnta=\z@
3805            \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
3806              Setting it to one}
3807          \fi
3808        \fi
3809        \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3810        \edef\MT@step{\number\@tempcnta\space}}%
```

\MT@auto Automatic expansion of the font? This new feature of pdfTeX 1.20 makes the
hz programme really usable. It must be either 'autoexpand' or empty (or '1000'
for older versions of pdfTeX).

```
3811      \let\MT@auto\@empty
3812      \ifMT@auto
3813        \MT@requires@pdftex4{%
```

We turn off automatic expansion if output mode is DVI.

```
3814          \ifnum\pdfoutput<\@ne
3815            \ifMT@opt@auto
3816              \MT@error{%
3817                Automatic font expansion only works for PDF output.\MessageBreak
3818                However, you are creating a DVI file}
3819              {If you have created expanded fonts instances, remove 'auto' from%
3820                \MessageBreak the package options. Otherwise, you have to switch
3821                off expansion\MessageBreak completely.}%
3822            \fi
3823            \MT@autofalse
3824          \else
3825            \def\MT@auto{autoexpand}%
3826          \fi
```

Also, if pdfTeX is too old.

```
3827        }{%
3828          \MT@error{%
3829            The pdftex version you are using is too old for\MessageBreak
3830            automatic font expansion}%
3831          {If you have created expanded fonts instances, remove 'auto' from\MessageBreak
3832            the package options. Otherwise, you have to switch off expansion\MessageBreak
3833            completely, or upgrade pdftex to version 1.20 or newer.}%
3834          \MT@autofalse
3835          \def\MT@auto{1000 }%
3836        }%
3837      \else
```

No automatic expansion.

```
3838        \MT@requires@pdftex4\relax{%
3839          \def\MT@auto{1000 }%
3840        }%
3841      \fi
```

Choose the appropriate macro for selected expansion.

```
3842      \ifMT@selected
3843        \let\MT@set@ex@codes\MT@set@ex@codes@s
3844      \else
```

```
3845        \let\MT@set@ex@codes\MT@set@ex@codes@n
3846      \fi
```

Filter out `stretch=0,shrink=0`, since it would result in a pdfTeX error.

```
3847      \ifnum\MT@stretch=\z@
3848        \ifnum\MT@shrink=\z@
3849          \MT@warning@nl{%
3850            Both the stretch and shrink limit are set to zero.\MessageBreak
3851            Disabling font expansion}%
3852          \MT@expansionfalse
3853        \fi
3854      \fi
3855    \fi
3856    \ifMT@expansion
3857      \edef\MT@active@features{\MT@active@features,ex}%
3858      \pdfadjustspacing\MT@ex@level
3859      \MT@info@nl{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
3860                 (level \number\MT@ex@level),\MessageBreak
3861                 stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3862                 step: \number\MT@step, \ifMT@selected\else non-\fi selected}%
```

`\MT@check@step`   Check whether `stretch` and `shrink` are multiples of `step`.

```
3863    \def\MT@check@step#1{%
3864      \@tempcnta=\csname MT@#1\endcsname
3865      \divide\@tempcnta \MT@step
3866      \multiply\@tempcnta \MT@step
3867      \ifnum\@tempcnta=\csname MT@#1\endcsname\else
3868        \MT@warning@nl{The #1 amount is not a multiple of step.\MessageBreak
3869                      The effective maximum #1 is \the\@tempcnta\space
3870                      (step \number\MT@step)}%
3871      \fi
3872    }%
3873    \MT@check@step{stretch}%
3874    \MT@check@step{shrink}%
3875    \MT@check@active@set{ex}%
```

Inside `\showhyphens`, font expansion should be disabled.

```
3876    \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3877      \color@begingroup\everypar{}\parfillskip\z@skip
3878      \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3879      \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}%
```

`\showhyphens`   I wonder why it's defined globally (in `ltfssbas.dtx`)?

```
3880    \gdef\showhyphens#1{\setbox0\vbox{%
3881      \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3882      \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
3883      \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}%
3884  \else
3885    \let\MT@expansion\relax
3886    \MT@info@nl{No font expansion}%
3887  \fi
3888 }
3889 \MT@requires@pdftex6{
```

`\MT@warn@lua`   Switch off the features that don't work with luaTeX.

```
3890 ⟨*lua⟩
3891  \def\MT@warn@lua#1{%
3892    \MT@error{The '#1' feature doesn't currently work\MessageBreak with luatex}
3893            {Use pdftex instead.}%
```

```
3894        \csname MT@#1false\endcsname
3895        \MT@let@nc{MT@#1}\relax
3896     }
3897  ⟨/lua⟩

3898  ⟨/package⟩
3899    \MT@addto@setup{%
3900  ⟨*package⟩
```

Tracking, spacing and kerning.

```
3901        \ifMT@tracking
3902  ⟨lua⟩        \MT@requires@luatex{\MT@warn@lua{tracking}}{%
3903            \edef\MT@active@features{\MT@active@features,tr}%
3904            \MT@info@nl{Tracking enabled}%
3905            \MT@check@active@set{tr}%
```

Enable protrusion for compensation at the line edges.

```
3906            \ifMT@protrusion\else\pdfprotrudechars\@ne\fi
3907  ⟨lua⟩        }%
3908        \else
3909          \let\MT@tracking\relax
3910          \MT@info@nl{No tracking}%
3911        \fi
3912        \ifMT@spacing
3913  ⟨lua⟩        \MT@requires@luatex{\MT@warn@lua{spacing}}{%
3914            \edef\MT@active@features{\MT@active@features,sp}%
3915            \pdfadjustinterwordglue\@ne
3916            \MT@info@nl{Adjustment of interword spacing enabled}%
3917            \MT@check@active@set{sp}%
3918  ⟨lua⟩        }%
3919        \else
3920          \let\MT@spacing\relax
3921          \MT@info@nl{No adjustment of interword spacing}%
3922        \fi
3923        \ifMT@kerning
3924  ⟨lua⟩        \MT@requires@luatex{\MT@warn@lua{kerning}}{%
3925            \edef\MT@active@features{\MT@active@features,kn}%
3926            \pdfprependkern\@ne
3927            \pdfappendkern\@ne
3928            \MT@info@nl{Adjustment of character kerning enabled}%
3929            \MT@check@active@set{kn}%
3930  ⟨lua⟩        }%
3931        \else
3932          \let\MT@kerning\relax
3933          \MT@info@nl{No adjustment of character kerning}%
3934        \fi
3935  ⟨/package⟩
```

\MT@warn@tracking@DVI   We issue a warning, when letterspacing in DVI mode, since it will probably not work. We also switch on protrusion if it isn't already, to compensate for the letterspacing kerns.

```
3936        \ifnum\pdfoutput<\@ne
3937          \def\MT@warn@tracking@DVI{%
3938            \MT@warning@nl{%
3939                You are using tracking/letterspacing in DVI mode.\MessageBreak
3940                This will probably not work, unless the post-\MessageBreak
3941                processing program (dvips, dvipdfm(x), ...) is\MessageBreak
3942                able to create the virtual fonts on the fly}%
3943            \MT@glet\MT@warn@tracking@DVI\relax
3944          }%
```

```
3945        \else
3946          \def\MT@warn@tracking@DVI{%
3947            \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
3948            \MT@glet\MT@warn@tracking@DVI\relax
3949          }%
3950        \fi
3951        \ifnum\MT@letterspace=\m@ne
3952          \let\MT@letterspace\MT@letterspace@default
3953        \else
3954          \MT@ls@too@large\MT@letterspace
3955        \fi
3956      }%
```

If pdfTeX is too old, we disable tracking, spacing and kerning, and throw an error
message.

```
3957  ⟨∗package⟩
3958  }{
3959    \MT@addto@setup{%
3960      \ifMT@tracking
3961        \MT@error{Tracking only works with pdftex version 1.40\MessageBreak
3962          or newer. Switching it off}{Upgrade pdftex.}%
3963      \else
3964        \MT@info@nl{No tracking (pdftex too old)}%
3965      \fi
3966      \ifMT@spacing
3967        \MT@error{Adjustment of interword spacing only works with\MessageBreak
3968          pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%
3969      \else
3970        \MT@info@nl{No adjustment of interword spacing (pdftex too old)}%
3971      \fi
3972      \ifMT@kerning
3973        \MT@error{Character kerning only works with\MessageBreak
3974          pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%
3975      \else
3976        \MT@info@nl{No adjustment of character kerning (pdftex too old)}%
3977      \fi
3978    }
3979  }
```

\DisableLigatures is only admissible in the preamble, therefore we can now
disable the corresponding macro, if it was never called.

```
3980  \MT@requires@pdftex5{
3981    \MT@addto@setup{%
3982      \ifMT@noligatures \else
3983        \let\MT@noligatures\relax
3984      \fi
3985    }
3986  }\relax
```

Remove the leading comma in \MT@active@features, and set the document switch
to true.

```
3987  \MT@addto@setup{%
3988    \ifx\MT@active@features\@empty \else
3989      \edef\MT@active@features{\expandafter\@gobble\MT@active@features}%
3990    \fi
3991    \MT@documenttrue
3992  }
```

\MT@set@babel@context   Interaction with babel.

```
3993 \def\MT@set@babel@context#1{%
3994   \MT@ifdefined@n@TF{MT@babel@#1}{%
3995     \MT@vinfo{*** Changing to language context '#1'\MessageBreak\on@line}%
3996     \expandafter\MT@exp@one@n\expandafter\microtypecontext
3997       \csname MT@babel@#1\endcsname
3998   }{%
3999     \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
4000   }%
4001 }
```

\MT@shorthandoff   Active characters can only be switched off if babel isn't loaded after microtype.

```
4002 \@ifpackageloaded{babel}{
4003   \def\MT@shorthandoff#1#2{%
4004     \MT@info@nl{Switching off #1 babel's active characters (#2)}%
4005     \shorthandoff{#2}}
4006 }{
4007   \def\MT@shorthandoff#1#2{%
4008     \MT@error{You must load 'babel' before '\MT@MT'}
4009             {Otherwise, '\MT@MT' cannot switch off #1 babel's\MessageBreak
4010              active characters.}}
4011 }
```

We patch the language switching commands to enable language-dependent setup.

```
4012 \MT@addto@setup{%
4013   \ifMT@babel
4014     \@ifpackageloaded{babel}{%
4015       \MT@info@nl{Redefining babel's language switching commands}%
4016       \let\MT@orig@select@language\select@language
4017       \def\select@language#1{%
4018         \MT@orig@select@language{#1}%
4019         \MT@set@babel@context{#1}%
4020       }%
4021       \let\MT@orig@foreign@language\foreign@language
4022       \def\foreign@language#1{%
4023         \MT@orig@foreign@language{#1}%
4024         \MT@set@babel@context{#1}%
4025       }%
4026       \ifMT@kerning
```

Disable French babel's active characters.

```
4027         \MT@if@false
4028         \MT@with@babel@and@T{french}  \MT@if@true
4029         \MT@with@babel@and@T{frenchb} \MT@if@true
4030         \MT@with@babel@and@T{francais}\MT@if@true
4031         \MT@with@babel@and@T{canadien}\MT@if@true
4032         \MT@with@babel@and@T{acadian} \MT@if@true
4033         \ifMT@if@\MT@shorthandoff{French}{:;!?}\fi
```

Disable Turkish babel's active characters.

```
4034         \MT@if@false
4035         \MT@with@babel@and@T{turkish} \MT@if@true
4036         \ifMT@if@\MT@shorthandoff{Turkish}{:!=}\fi
4037       \fi
```

In case babel was loaded before microtype:

```
4038       \MT@set@babel@context\languagename
4039     }{%
4040       \MT@warning@nl{You did not load the babel package.\MessageBreak
4041         The 'babel' option won't have any effect}%
4042     }%
```

```
4043    \fi
4044 }
```

Now we close the `\fi` from `\ifMT@draft`.

```
4045 \MT@addto@setup{\fi
```

Set up the current font, most likely the normal font. This has to come after all of the setup (including anything from the preamble) has been dealt with.

```
4046    \selectfont}
```

\MT@curr@file   This is the current file (hopefully with the correct extension).

```
4047 \edef\MT@curr@file{\jobname.tex}
```

Finally, execute the setup macro at the end of the preamble, and empty it (the **combine** class calls it repeatedly).

```
4048 ⟨/package⟩
4049 ⟨plain⟩\MT@requires@latex1{
4050 \AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@\@empty}
4051 ⟨plain⟩}\relax
```

Warning if `\nonfrenchspacing` is active, since space factors will be ignored with `\pdfadjustinterwordglue > 0`. Why 1500? Because some packages redefine `\frenchspacing`.[15] This has to be checked after the setup has taken place. There still will be a false warning if babel is loaded after microtype (without the babel option).

```
4052 ⟨*package⟩
4053 \MT@requires@pdftex6{
4054   \AtBeginDocument{%
4055     \ifMT@spacing
4056       \ifMT@babel \else
4057         \ifnum\sfcode`\. > 1500
4058           \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
4059             \MT@warning@nl{%
4060               \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
4061               interword spacing will disable it. You might want\MessageBreak
4062               to add `\@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
4063               to your preamble}%
4064           }%
4065         \fi
4066       \fi
4067     \fi
4068   }
4069 }\relax
4070 ⟨/package⟩
```

Restore catcodes.

```
4071 \MT@restore@catcodes
```

That was that.

```
4072 ⟨/package | letterspace⟩
```

15   Configuration files

Let's now write the font configuration files.

---

15   Cf. the c.t.t. thread '\frenchspacing with AMS packages and babel', started by Philipp Lehman on 16 August 2005, MID: ddtbajrob1@online.de

4073 ⟨∗config⟩
4074

15.1 Font sets

We first declare some sets in the main configuration file.

```
4075 ⟨∗m − t⟩
4076 %%% ----------------------------------------------------------------------
4077 %%% FONT SETS
4078
4079 \DeclareMicrotypeSet{all}
4080    { }
4081
4082 \DeclareMicrotypeSet{allmath}
4083    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
4084
4085 \DeclareMicrotypeSet{alltext}
4086    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
4087
4088 \DeclareMicrotypeSet{basicmath}
4089    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,OML,OMS},
4090      family   = {rm*,sf*},
4091      series   = {md*},
4092      size     = {normalsize,footnotesize,small,large}
4093    }
4094
4095 \DeclareMicrotypeSet{basictext}
4096    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5},
4097      family   = {rm*,sf*},
4098      series   = {md*},
4099      size     = {normalsize,footnotesize,small,large}
4100    }
4101
4102 \DeclareMicrotypeSet{smallcaps}
4103    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1},
4104      shape    = {sc*}
4105    }
4106
4107 \DeclareMicrotypeSet{footnotesize}
4108    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1},
4109      size     = {-small}
4110    }
4111
4112 \DeclareMicrotypeSet{scriptsize}
4113    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1},
4114      size     = {-footnotesize}
4115    }
4116
4117 \DeclareMicrotypeSet{normalfont}
4118    { font = */*/*/*/* }
4119
```

The default sets.

```
4120 %%% ----------------------------------------------------------------------
```

```
4121 %%% DEFAULT SETS
4122
4123 \DeclareMicrotypeSetDefault[protrusion]{alltext}
4124 \DeclareMicrotypeSetDefault[expansion] {basictext}
4125 \DeclareMicrotypeSetDefault[spacing]   {basictext}
4126 \DeclareMicrotypeSetDefault[kerning]   {alltext}
4127 \DeclareMicrotypeSetDefault[tracking]  {smallcaps}
4128
```

15.2 Font variants and aliases

```
4129 %%% -----------------------------------------------------------------------
4130 %%% FONT VARIANTS AND ALIASES
4131
```

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals):

```
4132 \DeclareMicrotypeVariants{x,j,w,a,d,0,1}
4133
```

Other candidates: 2 (proportional digits), e (engraved), f (Fraktur), g (small text), h (shadow), l (outline), n (informal), p (ornaments), r (roman), s (sans serif), t (typewriter). I've omitted them since they seem hardly be used and/or they are actually more than a variant, i. e., they shouldn't share a file.

Fonts that are 'the same': The Latin Modern fonts, the virtual fonts from the ae and zefonts, and the eco and hfoldsty packages (oldstyle numerals) all inherit the (basic) settings from Computer Modern Roman. Some of them are in part overwritten later.

```
4134 \DeclareMicrotypeAlias{lmr} {cmr}  % lmodern
4135 \DeclareMicrotypeAlias{aer} {cmr}  % ae
4136 \DeclareMicrotypeAlias{zer} {cmr}  % zefonts
4137 \DeclareMicrotypeAlias{cmor}{cmr}  % eco
4138 \DeclareMicrotypeAlias{hfor}{cmr}  % hfoldsty
```

The packages pxfonts and txfonts fonts inherit Palatino and Times settings respectively, also the TEX Gyre fonts Pagella and Termes (formerly: qfonts).

```
4139 \DeclareMicrotypeAlias{pxr} {ppl}  % pxfonts
4140 \DeclareMicrotypeAlias{qpl} {ppl}  % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)
```

The 'FPL Neu' fonts, a 're-implementation' of Palatino.

```
4141 \DeclareMicrotypeAlias{fp9x}{pplx} % FPL Neu
4142 \DeclareMicrotypeAlias{fp9j}{pplj} %    "
4143 \DeclareMicrotypeAlias{txr} {ptm}  % txfonts
4144 \DeclareMicrotypeAlias{qtm} {ptm}  % TeX Gyre Termes (formerly: qfonts/QuasiTimes)
```

More Times variants, to be checked: pns, mns (TimesNewRomanPS); mnt (TimesNewRomanMT, TimesNRSevenMT), mtm (TimesSmallTextMT); pte (TimesEuropa); ptt (TimesTen); TimesEighteen; TimesModernEF.
    The eulervm package virtually extends the Euler fonts.

```
4145 \DeclareMicrotypeAlias{zeur}{eur}  % Euler VM
4146 \DeclareMicrotypeAlias{zeus}{eus}  %    "
```

MicroPress's Charter version (chmath).

```
4147 \DeclareMicrotypeAlias{chr} {bch}  % CH Math
```

The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

```
4148 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
4149 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
4150 \DeclareMicrotypeAlias{ulg} {blg}  % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

Euro symbol fonts, to save some files.

```
4151 \DeclareMicrotypeAlias{zpeus}   {zpeu}    % Adobe Euro sans -> serif
4152 \DeclareMicrotypeAlias{eurosans}{zpeu}    % Adobe Euro sans -> serif
4153 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
4154
```

15.3 Interaction with babel

Contexts that are to be set when switching to a language.

```
4155 %%% ----------------------------------------------------------------------
4156 %%% INTERACTION WITH THE 'babel' PACKAGE
4157
4158 \DeclareMicrotypeBabelHook
4159    {english,UKenglish,british,USenglish,american}
4160    {kerning=, spacing=nonfrench}
4161
4162 \DeclareMicrotypeBabelHook
4163    {french,francais,acadian,canadien}
4164    {kerning=french, spacing=}
4165
4166 \DeclareMicrotypeBabelHook
4167    {turkish}
4168    {kerning=turkish, spacing=}
4169
```

15.4 Note on admissible characters

All printable ASCII characters are allowed in the settings, with the following exceptions (on the left hand side, the replacements on the right):

```
\ : \textbackslash
{ : \textbraceleft
} : \textbraceright
^ : \textasciicircum
% : \%
# : \#
```

Comma and equal sign must be guarded with braces ('{,}', '{=}') to keep keyval happy.

Character commands are allowed as far as they have been defined in the proper LATEX way, that is, when they have been assigned a slot in the font encoding with \DeclareTextSymbol or \DeclareTextComposite. Characters defined via \chardef are also possible.

Ligatures and \mathchardefed symbols have to be specified numerically. Of course, numerical identification is possible in any other case, too.

8-bit characters are also admissible, provided they have been declared in the input encoding file. They should, however, only be used in private configuration files, where the proper input encoding is guaranteed, or else in combination with the 'inputenc' key.

15.5 Character inheritance

First the lists of inheriting characters. We only declare those characters that are the same on *both* sides, i. e., not Œ for O.

```
4170 ⟨/m − t⟩
4171 ⟨∗m − t | zpeu | mvs⟩
4172 %%% -------------------------------------------------------------------------
4173 %%% CHARACTER INHERITANCE
4174
4175 ⟨/m − t | zpeu | mvs⟩
4176 ⟨∗m − t⟩
```

15.5.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 ('fi' ligature), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```
4177 \DeclareCharacterInheritance
4178   { encoding = OT1 }
4179   { f = {011}, % ff
4180     i = {\i},
4181     j = {\j},
4182     O = {\O},
4183     o = {\o}
4184   }
4185
```

15.5.2 T1

Candidates here: 028 ('fi'), 029 ('fl'), 030 ('ffi'), 031 ('ffl'), 156 ('IJ' ligature, since LaTeX 2005/12/01 accessible as \IJ), 188 ('ij', \ij), Æ, æ, Œ, œ.

```
4186 \DeclareCharacterInheritance
4187   { encoding = T1 }
4188   { A = {\'A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
4189     a = {\'a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
4190     C = {\'C,\c C,\v C},
4191     c = {\'c,\c c,\v c},
4192     D = {\v D,\DH},
4193     d = {\v d,\dj},
4194     E = {\'E,\'E,\^E,\"E,\k E,\v E},
4195     e = {\'e,\'e,\^e,\"e,\k e,\v e},
4196     f = {027}, % ff
4197     G = {\u G},
4198     g = {\u g},
4199     I = {\'I,\'I,\^I,\"I,\.I},
4200     i = {\'i,\'i,\^i,\"i,\i},
4201     j = {\j},
4202     L = {\L,\'L,\v L},
4203     l = {\l,\'l,\v l},
4204     N = {\'N,\~N,\v N},
4205     n = {\'n,\~n,\v n},
4206     O = {\O,\'O,\'O,\^O,\~O,\"O,\H O},
4207     o = {\o,\'o,\'o,\^o,\~o,\"o,\H o},
4208     R = {\'R,\v R},
4209     r = {\'r,\v r},
4210     S = {\'S,\c S,\v S,\SS},
4211     s = {\'s,\c s,\v s},
4212     T = {\c T,\v T},
```

```
4213        t = {\c t,\v t},
4214        U = {\'U,\'U,\^U,\"U,\H U,\r U},
4215        u = {\'u,\'u,\^u,\"u,\H u,\r u},
4216        Y = {\'Y,\"Y},
4217        y = {\'y,\"y},
4218        Z = {\'Z,\.Z,\v Z},
4219        z = {\'z,\.z,\v z}
```

The 'soft hyphen' often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```
4220 %      - = {127},
4221    }
4222
```

15.5.3 LY1

More characters: 008 ('fl'), 012 ('fi'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```
4223 \DeclareCharacterInheritance
4224    { encoding = LY1 }
4225    { A = {\'A,\'A,\^A,\~A,\"A,\r A},
4226      a = {\'a,\'a,\^a,\~a,\"a,\r a},
4227      C = {\c C},
4228      c = {\c c},
4229      D = {\DH},
4230      E = {\'E,\'E,\^E,\"E},
4231      e = {\'e,\'e,\^e,\"e},
4232      f = {011}, % ff
4233      I = {\'I,\'I,\^I,\"I},
4234      i = {\'i,\'i,\^i,\"i,\i},
4235      L = {\L},
4236      l = {\l},
4237      N = {\~N},
4238      n = {\~n},
4239      O = {\'O,\'O,\^O,\~O,\"O,\O},
4240      o = {\'o,\'o,\^o,\~o,\"o,\o},
4241      S = {\v S},
4242      s = {\v s},
4243      U = {\'U,\'U,\^U,\"U},
4244      u = {\'u,\'u,\^u,\"u},
4245      Y = {\'Y,\"Y},
4246      y = {\'y,\"y},
4247      Z = {\v Z},
4248      z = {\v z}
4249    }
4250
```

15.5.4 OT4

The Polish OT1 extension. More interesting characters here: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```
4251 \DeclareCharacterInheritance
4252    { encoding = OT4 }
4253    { A = {\k A},
4254      a = {\k a},
4255      C = {\'C},
4256      c = {\'c},
4257      E = {\k E},
4258      e = {\k e},
```

```
4259        f = {011}, % ff
4260        i = {\i},
4261        j = {\j},
4262        L = {\L},
4263        l = {\l},
4264        N = {\'N},
4265        n = {\'n},
4266        O = {\O,\'O},
4267        o = {\o,\'o},
4268        S = {\'S},
4269        s = {\'s},
4270        Z = {\'Z,\.Z},
4271        z = {\'z,\.z}
4272      }
4273
```

15.5.5 QX

The Central European QX encoding.[16] Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```
4274  \DeclareCharacterInheritance
4275    { encoding = QX }
4276    { A = {\`A,\'A,\^A,\~A,\"A,\k A,\AA},
4277      a = {\`a,\'a,\^a,\~a,\"a,\k a,\aa},
4278      C = {\'C,\c C},
4279      c = {\'c,\c c},
4280      D = {\DH},
4281      E = {\`E,\'E,\^E,\"E,\k E},
4282      e = {\`e,\'e,\^e,\"e,\k e},
4283      f = {011}, % ff
4284      I = {\`I,\'I,\^I,\"I,\k I},
4285      i = {\`i,\'i,\^i,\"i,\k i,\i},
4286      j = {\j},
4287      L = {\L},
4288      l = {\l},
4289      N = {\'N,\~N},
4290      n = {\'n,\~n},
4291      O = {\O,\`O,\'O,\^O,\~O,\"O},
4292      o = {\o,\`o,\'o,\^o,\~o,\"o},
```

The Rumanian \textcommabelow accents are actually replacements for the \c variants, which had previously (and erroneously[17]) been included in QX encoding. They are still kept for backwards compatibility.

```
4293      S = {\'S,\c S,\textcommabelow S,\v S},
4294      s = {\'s,\c s,\textcommabelow s,\v s},
4295      T = {\c T,\textcommabelow T},
4296      t = {\c t,\textcommabelow t},
4297      U = {\`U,\'U,\^U,\"U,\k U},
4298      u = {\`u,\'u,\^u,\"u,\k u},
4299      Y = {\'Y,\"Y},
4300      y = {\'y,\"y},
4301      Z = {\'Z,\.Z,\v Z},
4302      z = {\'z,\.z,\v z},
4303      . = \textellipsis
4304    }
4305
```

---

16 Contributed by *Maciej Eder*.
17 Cf. http://tug.org/pipermail/tex-live/2008-August/017204.html

15.5.6 T5

The Vietnamese encoding T5. It is so crowded with accented and double-accented characters that there is no room for any ligatures.

```
4306 \DeclareCharacterInheritance
4307   { encoding = T5 }
4308   { A = {\`A,\'A,\~A,\h A,\d A,\^A,\u A,
4309         \`\Acircumflex,\'\Acircumflex,\~\Acircumflex,\h\Acircumflex,\d\Acircumflex,
4310         \`\Abreve,\'\Abreve,\~\Abreve,\h\Abreve,\d\Abreve},
4311     a = {\`a,\'a,\~a,\h a,\d a,\^a,\u a,
4312         \`\acircumflex,\'\acircumflex,\~\acircumflex,\h\acircumflex,\d\acircumflex,
4313         \`\abreve,\'\abreve,\~\abreve,\h\abreve,\d\abreve},
4314     D = {\DJ},
4315     d = {\dj},
4316     E = {\`E,\'E,\~E,\h E,\d E,\^E,
4317         \`\Ecircumflex,\'\Ecircumflex,\~\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
4318     e = {\`e,\'e,\~e,\h e,\d e,\^e,
4319         \`\ecircumflex,\'\ecircumflex,\~\ecircumflex,\h\ecircumflex,\d\ecircumflex},
4320     I = {\`I,\'I,\~I,\h I,\d I},
4321     i = {\`i,\'i,\~i,\h i,\d i,\i},
4322     O = {\`O,\'O,\~O,\h O,\d O,\^O,\horn O,
4323         \`\Ocircumflex,\'\Ocircumflex,\~\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
4324         \`\Ohorn,\'\Ohorn,\~\Ohorn,\h\Ohorn,\d\Ohorn},
4325     o = {\`o,\'o,\~o,\h o,\d o,\^o,\horn o,
4326         \`\ocircumflex,\'\ocircumflex,\~\ocircumflex,\h\ocircumflex,\d\ocircumflex,
4327         \`\ohorn,\'\ohorn,\~\ohorn,\h\ohorn,\d\ohorn},
4328     U = {\`U,\'U,\~U,\h U,\d U,\horn U,
4329         \`\Uhorn,\'\Uhorn,\~\Uhorn,\h\Uhorn,\d\Uhorn},
4330     u = {\`u,\'u,\~u,\h u,\d u,\horn u,
4331         \`\uhorn,\'\uhorn,\~\uhorn,\h\uhorn,\d\uhorn},
4332     Y = {\`Y,\'Y,\~Y,\h Y,\d Y},
4333     y = {\`y,\'y,\~y,\h y,\d y}
4334   }
4335
4336 ⟨/m − t⟩
```

15.5.7 Euro symbols

Make Euro symbols settings simpler.

```
4337 ⟨*zpeu⟩
4338 \DeclareCharacterInheritance
4339   { encoding = U,
4340     family   = {zpeu,zpeus,eurosans} }
4341   { E = 128 }
4342
4343 ⟨/zpeu⟩
4344 ⟨*mvs⟩
4345 \DeclareCharacterInheritance
4346   { encoding = OT1,
4347     family   = mvs }
4348   { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
4349
```

Since 2006/05/11 (that is, one week after I've added these settings, after the package had been dormant for six years), marvosym's encoding is (correctly) U instead of OT1.

```
4350 \DeclareCharacterInheritance
4351   { encoding = U,
```

```
4352      family   = mvs }
4353    { 164 = {099,100,101} }
4354
4355  ⟨/mvs⟩
```

15.6 Tracking

By default, we only disable the 'f*' ligatures, for those fonts that have any. Thus, ligatures and especially kerning for all other characters will be retained.

```
4356  ⟨∗m − t⟩
4357  %%% ------------------------------------------------------------------------
4358  %%% TRACKING/LETTERSPACING
4359
4360  \SetTracking
4361    [ name          = default,
4362      no ligatures = {f} ]
4363    { encoding      = {OT1,T1,T2A,LY1,OT4,QX} }
4364    { }
4365
```

15.7 Font expansion

These are Hàn Thế Thành's original expansion settings. They are used for all fonts (until somebody shows mercy and creates font-specific settings).

```
4366  %%% ------------------------------------------------------------------------
4367  %%% EXPANSION
4368
4369  \SetExpansion
4370    [ name      = default      ]
4371    { encoding = {OT1,OT4,QX,T1,LY1} }
4372    {
4373      A = 500,     a = 700,
4374    \AE = 500,   \ae = 700,
4375      B = 700,     b = 700,
4376      C = 700,     c = 700,
4377      D = 500,     d = 700,
4378      E = 700,     e = 700,
4379      F = 700,
4380      G = 500,     g = 700,
4381      H = 700,     h = 700,
4382      K = 700,     k = 700,
4383      M = 700,     m = 700,
4384      N = 700,     n = 700,
4385      O = 500,     o = 700,
4386    \OE = 500,   \oe = 700,
4387      P = 700,     p = 700,
4388      Q = 500,     q = 700,
4389      R = 700,
4390      S = 700,     s = 700,
4391      U = 700,     u = 700,
4392      W = 700,     w = 700,
4393      Z = 700,     z = 700,
4394      2 = 700,
4395      3 = 700,
4396      6 = 700,
4397      8 = 700,
4398      9 = 700
```

```
4399     }
4400
```

Settings for Cyrillic T2A encoding.[18]

```
4401  \SetExpansion
4402    [ name      = T2A ]
4403    { encoding = T2A }
4404    {
4405      A = 500,      a = 700,
4406      B = 700,      b = 700,
4407      C = 700,      c = 700,
4408      D = 500,      d = 700,
4409      E = 700,      e = 700,
4410      F = 700,
4411      G = 500,      g = 700,
4412      H = 700,      h = 700,
4413      K = 700,      k = 700,
4414      M = 700,      m = 700,
4415      N = 700,      n = 700,
4416      O = 500,      o = 700,
4417      P = 700,      p = 700,
4418      Q = 500,      q = 700,
4419      R = 700,
4420      S = 700,      s = 700,
4421      U = 700,      u = 700,
4422      W = 700,      w = 700,
4423      Z = 700,      z = 700,
4424      2 = 700,
4425      3 = 700,
4426      6 = 700,
4427      8 = 700,
4428      9 = 700,
4429      \CYRA = 500,      \cyra = 700,
4430      \CYRB = 700,      \cyrb = 700,
4431      \CYRV = 700,      \cyrv = 700,
4432      \CYRG = 700,      \cyrg = 700,
4433      \CYRD = 700,      \cyrd = 700,
4434      \CYRE = 700,      \cyre = 700,
4435      \CYRZH = 700,     \cyrzh = 700,
4436      \CYRZ = 700,      \cyrz = 700,
4437      \CYRI = 700,      \cyri = 700,
4438      \CYRISHRT = 700, \cyrishrt = 700,
4439      \CYRK = 700,      \cyrk = 700,
4440      \CYRL = 700,      \cyrl = 700,
4441      \CYRM = 700,      \cyrm = 700,
4442      \CYRN = 700,      \cyrn = 700,
4443      \CYRO = 500,      \cyro = 700,
4444      \CYRP = 700,      \cyrp = 700,
4445      \CYRR = 700,      \cyrr = 700,
4446      \CYRS = 700,      \cyrs = 700,
4447      \CYRT = 700,      \cyrt = 700,
4448      \CYRU = 700,      \cyru = 700,
4449      \CYRF = 700,      \cyrf = 700,
4450      \CYRH = 700,      \cyrh = 700,
4451      \CYRC = 700,      \cyrc = 700,
4452      \CYRCH = 700,     \cyrch = 700,
4453      \CYRSH = 700,     \cyrsh = 700,
4454      \CYRSHCH = 700,  \cyrshch = 700,
```

---

18   Contributed by *Karl Karlsson*.

```
4455      \CYRHRDSN = 700, \cyrhrdsn = 700,
4456      \CYRERY = 700,   \cyrery = 700,
4457      \CYRSFTSN = 700, \cyrsftsn = 700,
4458      \CYREREV = 700,  \cyrerev = 700,
4459      \CYRYU = 700,    \cyryu = 700,
4460      \CYRYA = 700,    \cyrya = 700
4461    }
4462
```

T5 encoding does not contain \AE, \ae, \OE and \oe.

```
4463 \SetExpansion
4464    [ name      = T5 ]
4465    { encoding = T5 }
4466    {
4467      A = 500,      a = 700,
4468      B = 700,      b = 700,
4469      C = 700,      c = 700,
4470      D = 500,      d = 700,
4471      E = 700,      e = 700,
4472      F = 700,
4473      G = 500,      g = 700,
4474      H = 700,      h = 700,
4475      K = 700,      k = 700,
4476      M = 700,      m = 700,
4477      N = 700,      n = 700,
4478      O = 500,      o = 700,
4479      P = 700,      p = 700,
4480      Q = 500,      q = 700,
4481      R = 700,
4482      S = 700,      s = 700,
4483      U = 700,      u = 700,
4484      W = 700,      w = 700,
4485      Z = 700,      z = 700,
4486      2 = 700,
4487      3 = 700,
4488      6 = 700,
4489      8 = 700,
4490      9 = 700
4491    }
4492
4493 ⟨/m − t⟩
```

15.8 Character protrusion

```
4494 %%% --------------------------------------------------------------------
4495 %%% PROTRUSION
4496
```

For future historians, Hàn Thế Thành's original settings (from `protcode.tex`, converted to microtype notation).

```
\SetProtrusion
   [ name      = thanh ]
   { encoding = OT1 }
   {
     A = {50,50},
     F = {  ,50},
     J = {50,  },
     K = {  ,50},
     L = {  ,50},
```

```
        T = {50,50},
        V = {50,50},
        W = {50,50},
        X = {50,50},
        Y = {50,50},
        k = {  ,50},
        r = {  ,50},
        t = {  ,50},
        v = {50,50},
        w = {50,50},
        x = {50,50},
        y = {50,50},
        . = { ,700},    {,}= { ,700},
        : = { ,500},     ; = { ,500},
        ! = { ,200},     ? = { ,200},
        ( = {50, },     ) = {  ,50},
        - = { ,700},
        \textendash       = { ,300},     \textemdash       = { ,200},
        \textquoteleft    = {700, },     \textquoteright   = { ,700},
        \textquotedblleft  = {500, },     \textquotedblright = { ,500}
      }
```

15.8.1 Normal

The default settings always use the most moderate value.

```
4497 ⟨∗cfg − t⟩
4498 \SetProtrusion
4499 ⟨m − t⟩   [ name      = default ]
```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

```
4500 ⟨bch⟩   [ name      = bch-default ]
```

- Bitstream Letter Gothic (blg)

```
4501 ⟨blg⟩   [ name      = blg-default ]
```

- Computer Modern Roman (cmr)

```
4502 ⟨cmr⟩   [ name      = cmr-default ]
```

- Adobe Garamond (pad, padx, padj)

```
4503 ⟨pad⟩   [ name      = pad-default ]
```

- Minion[19] (pmnx, pmnj)

```
4504 ⟨pmn⟩   [ name      = pmnj-default ]
```

- Palatino (ppl, pplx, pplj)

```
4505 ⟨ppl⟩   [ name      = ppl-default ]
```

- Times (ptm, ptmx, ptmj)

```
4506 ⟨ptm⟩   [ name      = ptm-default ]
```

- URW Garamond (ugm)

---

19 Contributed by *Harald Harders* and *Karl Karlsson*.

```
4507 ⟨ugm⟩    [ name      = ugm-default ]
4508 ⟨m − t | cmr | pmn⟩    { }
4509 ⟨bch | blg | pad | ugm⟩    { encoding = OT1,
4510 ⟨ppl | ptm⟩    { encoding = {OT1,OT4},
4511 ⟨bch⟩       family   = bch }
4512 ⟨blg⟩       family   = blg }
4513 ⟨pad⟩       family   = {pad,padx,padj} }
4514 ⟨ppl⟩       family   = {ppl,pplx,pplj} }
4515 ⟨ptm⟩       family   = {ptm,ptmx,ptmj} }
4516 ⟨ugm⟩       family   = ugm }
4517    {
4518 ⟨m − t | bch | blg | cmr | pad | pmn | ppl | ptm⟩      A = {50,50},
4519 ⟨ugm⟩       A = {50,100},
4520 ⟨pad | ptm⟩    \AE = {50,  },
4521 ⟨ugm⟩    \AE = {150,50},
4522 ⟨ugm⟩       B = {  ,50},
4523 ⟨bch | pad | pmn | ugm⟩       C = {50,  },
4524 ⟨bch | pad | pmn⟩       D = {  ,50},
4525 ⟨ugm⟩       D = {  ,70},
4526 ⟨ugm⟩       E = {  ,50},
4527 ⟨m − t | bch | cmr | pad | pmn | ptm⟩      F = {  ,50},
4528 ⟨ugm⟩       F = {  ,70},
4529 ⟨bch | pad | pmn⟩       G = {50,  },
4530 ⟨ugm⟩       G = {50,50},
4531 ⟨blg⟩       I = {150,150},
4532 ⟨m − t | cmr | pad | pmn | ppl | ptm | ugm⟩      J = {50,  },
4533 ⟨bch | blg⟩       J = {100,  },
4534 ⟨!blg⟩       K = {  ,50},
4535 ⟨blg⟩       K = {50,  },
4536 ⟨m − t | bch | cmr | pad | pmn | ppl⟩      L = {  ,50},
4537 ⟨blg⟩       L = {  ,150},
4538 ⟨ptm⟩       L = {  ,80},
4539 ⟨ugm⟩       L = {  ,120},
4540 ⟨bch | pad | pmn | ugm⟩      O = {50,50},
4541 ⟨pad⟩    \OE = {50,  },
4542 ⟨ugm⟩    \OE = {50,50},
4543 ⟨blg⟩       P = {  ,100},
4544 ⟨ugm⟩       P = {  ,50},
4545 ⟨bch | pad | pmn⟩       Q = {50,70},
4546 ⟨ugm⟩       Q = {50,50},
4547 ⟨bch⟩       R = {  ,50},
4548 ⟨ugm⟩       R = {  ,70},
4549 ⟨m − t | bch | cmr | pad | pmn | ppl | ptm⟩      T = {50,50},
4550 ⟨blg⟩       T = {100,100},
4551 ⟨ugm⟩       T = {70,70},
4552 ⟨m − t | bch | cmr | pad | pmn | ppl | ptm⟩      V = {50,50},
4553 ⟨blg | ugm⟩       V = {70,70},
4554 ⟨m − t | bch | cmr | pad | pmn | ppl | ptm⟩      W = {50,50},
4555 ⟨ugm⟩       W = {70,70},
4556 ⟨m − t | bch | cmr | pad | pmn | ppl | ptm⟩      X = {50,50},
4557 ⟨ugm⟩       X = {50,70},
4558 ⟨m − t | bch | cmr | pad | pmn | ppl⟩      Y = {50,50},
4559 ⟨blg | ptm | ugm⟩       Y = {80,80},
4560 ⟨ugm⟩       Z = {50,50},
4561 ⟨blg⟩       f = {150,100},
4562 ⟨blg⟩       i = {150,150},
4563 ⟨blg⟩       j = {100,100},
4564 ⟨m − t | bch | cmr | pad | pmn | ppl | ptm⟩      k = {  ,50},
4565 ⟨ugm⟩       k = {  ,70},
4566 ⟨blg⟩       l = {150,150},
```

```
4567 ⟨pmn⟩        l = {  ,-50},
4568 ⟨pad | ppl⟩      p = {50,50},
4569 ⟨ugm⟩      p = {  ,50},
4570 ⟨pad | ppl⟩      q = {50,  },
4571 ⟨!blg⟩      r = {  ,50},
4572 ⟨blg⟩      r = {100, 80},
4573 ⟨cmr | pad | pmn⟩      t = {  ,70},
4574 ⟨bch⟩      t = {  ,50},
4575 ⟨blg⟩      t = {150, 80},
4576 ⟨ugm⟩      t = {  ,100},
4577 ⟨m − t | bch | cmr | pad | pmn | ppl | ptm⟩      v = {50,50},
4578 ⟨blg⟩      v = {100,100},
4579 ⟨ugm⟩      v = {50,70},
4580 ⟨m − t | bch | cmr | pad | pmn | ppl | ptm⟩      w = {50,50},
4581 ⟨ugm⟩      w = {50,70},
4582 ⟨!blg⟩      x = {50,50},
4583 ⟨blg⟩      x = {100,100},
4584 ⟨m − t | bch | pad | pmn⟩      y = {  ,50},
4585 ⟨blg⟩      y = { 50,100},
4586 ⟨cmr | ppl | ptm⟩      y = {50,70},
4587 ⟨ugm⟩      y = {  ,70},

4588 ⟨cmr⟩      0 = {  ,50},
4589 ⟨m − t⟩      1 = {50,50},
4590 ⟨bch | blg | pad | ptm | ugm⟩      1 = {150,150},
4591 ⟨cmr⟩      1 = {100,200},
4592 ⟨pmn⟩      1 = {  ,50},
4593 ⟨ppl⟩      1 = {100,100},
4594 ⟨bch | cmr | pad | ugm⟩      2 = {50,50},
4595 ⟨blg⟩      2 = {  ,100},
4596 ⟨bch | pmn⟩      3 = {50,  },
4597 ⟨cmr | pad | ugm⟩      3 = {50,50},
4598 ⟨blg⟩      3 = {100,  },
4599 ⟨m − t | pad⟩      4 = {50,50},
4600 ⟨bch⟩      4 = {100,50},
4601 ⟨blg⟩      4 = {100,  },
4602 ⟨cmr | ugm⟩      4 = {70,70},
4603 ⟨pmn⟩      4 = {50,  },
4604 ⟨ptm⟩      4 = {70,  },
4605 ⟨cmr⟩      5 = {  ,50},
4606 ⟨pad⟩      5 = {50,50},
4607 ⟨bch⟩      6 = {50,  },
4608 ⟨cmr⟩      6 = {  ,50},
4609 ⟨pad⟩      6 = {50,50},
4610 ⟨m − t⟩      7 = {50,50},
4611 ⟨bch | pad | pmn | ugm⟩      7 = {50,80},
4612 ⟨blg⟩      7 = {100,100},
4613 ⟨cmr | ptm⟩      7 = {50,100},
4614 ⟨ppl⟩      7 = {  ,50},
4615 ⟨cmr⟩      8 = {  ,50},
4616 ⟨bch | pad⟩      9 = {50,50},
4617 ⟨cmr⟩      9 = {  ,50},
4618 ⟨m − t | cmr | pad | pmn | ppl | ptm | ugm⟩      . = { ,700},
4619 ⟨bch⟩      . = { ,600},
4620 ⟨blg⟩      . = {400,500},
4621 ⟨!blg⟩      {,}= { ,500},
4622 ⟨blg⟩      {,}= {300,400},
4623 ⟨m − t | cmr | pad | pmn | ppl | ptm | ugm⟩      : = { ,500},
4624 ⟨bch⟩      : = { ,400},
4625 ⟨blg⟩      : = {300,400},
```

```
4626 ⟨m − t | bch | pad | pmn | ptm⟩      ; = {  ,300},
4627 ⟨blg⟩      ; = {200,300},
4628 ⟨cmr | ppl⟩      ; = {  ,500},
4629 ⟨ugm⟩      ; = {  ,400},
4630 ⟨!blg⟩      ! = {  ,100},
4631 ⟨blg⟩      ! = {200,200},
4632 ⟨m − t | pad | pmn | ptm⟩      ? = {  ,100},
4633 ⟨bch | cmr | ppl | ugm⟩      ? = {  ,200},
4634 ⟨blg⟩      ? = {150,150},
4635 ⟨pmn⟩      " = {300,300},
4636 ⟨m − t | bch | cmr | pad | pmn | ppl⟩      @ = {50,50},
4637 ⟨ptm⟩      @ = {100,100},
4638 ⟨m − t | bch | blg | cmr | pad | pmn | ppl | ptm⟩      ~ = {200,250},
4639 ⟨ugm⟩      ~ = {300,350},
4640 ⟨pad | ppl | ptm⟩      & = {50,100},
4641 ⟨ugm⟩      & = {  ,100},
4642 ⟨m − t | cmr | pad | pmn⟩      \% = {50,50},
4643 ⟨bch⟩      \% = {  ,50},
4644 ⟨ppl | ptm⟩      \% = {100,100},
4645 ⟨ugm⟩      \% = {50,100},
4646 ⟨blg⟩      \# = {100,100},
4647 ⟨m − t | ppl | ptm | ugm⟩      * = {200,200},
4648 ⟨bch | pmn⟩      * = {200,300},
4649 ⟨blg⟩      * = {150,200},
4650 ⟨cmr | pad⟩      * = {300,300},
4651 ⟨m − t | cmr | ppl | ptm⟩      + = {250,250},
4652 ⟨bch⟩      + = {150,250},
4653 ⟨pad⟩      + = {300,300},
4654 ⟨blg | pmn⟩      + = {150,200},
4655 ⟨ugm⟩      + = {250,300},
4656 ⟨blg | ugm⟩      {=}= {200,200},
4657 ⟨m − t | pad | pmn | ptm⟩      ( = {100,  },      ) = {  ,200},
4658 ⟨bch | ugm⟩      ( = {200,  },      ) = {  ,200},
4659 ⟨cmr | blg⟩      ( = {300,  },      ) = {  ,300},
4660 ⟨ppl⟩      ( = {100,  },      ) = {  ,300},
4661 ⟨bch | pmn⟩      [ = {100,  },      ] = {  ,100},
4662 ⟨blg⟩      [ = {300,100},      ] = {  ,300},

4663 ⟨m − t | pad | pmn | ptm⟩      / = {100,200},
4664 ⟨bch⟩      / = {  ,200},
4665 ⟨blg⟩      / = {300,300},
4666 ⟨cmr | ppl⟩      / = {200,300},
4667 ⟨ugm⟩      / = {100,300},
4668 ⟨m − t | ptm⟩      - = {500,500},
4669 ⟨bch | cmr | ppl⟩      - = {400,500},
4670 ⟨blg⟩      - = {300,400},
4671 ⟨pad⟩      - = {300,500},
4672 ⟨pmn⟩      - = {200,400},
4673 ⟨ugm⟩      - = {500,600},
4674 ⟨blg⟩      < = {200,100},      > = {100,200},
4675 ⟨blg⟩      _ = {150,250},
4676 ⟨blg⟩      | = {250,250},
4677 ⟨m − t | pmn⟩      \textendash      = {200,200},      \textemdash      = {150,150},
4678 ⟨bch⟩      \textendash      = {200,300},      \textemdash      = {150,250},
4679 ⟨cmr⟩      \textendash      = {400,300},      \textemdash      = {300,200},
4680 ⟨pad | ppl | ptm⟩      \textendash      = {300,300},      \textemdash      = {200,200},
4681 ⟨ugm⟩      \textendash      = {250,300},      \textemdash      = {250,250},
```

Why settings for left *and* right quotes? Because in some languages they might be
used like that (see the csquotes package for examples).

```
4682 ⟨m − t | bch | pmn⟩      \textquoteleft    = {300,400},   \textquoteright    = {300,400},
4683 ⟨blg⟩     \textquoteleft    = {400,600},   \textquoteright    = {400,600},
4684 ⟨cmr⟩     \textquoteleft    = {500,700},   \textquoteright    = {500,600},
4685 ⟨pad | ppl⟩     \textquoteleft    = {500,700},   \textquoteright    = {500,700},
4686 ⟨ptm⟩     \textquoteleft    = {500,500},   \textquoteright    = {300,500},
4687 ⟨ugm⟩     \textquoteleft    = {300,600},   \textquoteright    = {300,600},
4688 ⟨m − t | bch | pmn⟩      \textquotedblleft = {300,300},   \textquotedblright = {300,300}
4689 ⟨blg⟩     \textquotedblright = {300,400}
4690 ⟨cmr⟩     \textquotedblleft = {500,300},   \textquotedblright = {200,600}
4691 ⟨pad | ppl | ptm⟩      \textquotedblleft = {300,400},   \textquotedblright = {300,400}
4692 ⟨ugm⟩     \textquotedblleft = {400,400},   \textquotedblright = {400,400}
4693    }
4694
```

Greek uppercase letters are in OT1 encoding only.

```
4695 ⟨∗m − t | cmr | pmn⟩
4696 \SetProtrusion
4697 ⟨m − t⟩   [ name      = OT1-default,
4698 ⟨cmr⟩   [ name      = cmr-OT1,
4699 ⟨pmn⟩   [ name      = pmnj-OT1,
4700 ⟨m − t⟩     load     = default ]
4701 ⟨cmr⟩     load     = cmr-default ]
4702 ⟨pmn⟩     load     = pmnj-default ]
4703 ⟨m − t⟩   { encoding = OT1 }
4704 ⟨cmr⟩   { encoding = {OT1,OT4},
4705 ⟨pmn⟩   { encoding = OT1,
4706 ⟨cmr⟩     family   = cmr  }
4707 ⟨pmn⟩     family   = pmnj }
4708    {
4709 ⟨m − t | cmr⟩     \AE = {50,   },
4710 ⟨pmn⟩     \OE = {50,  }
4711 ⟨∗cmr⟩
4712    "00 = {   ,150}, % \Gamma
4713    "01 = {100,100}, % \Delta
4714    "02 = { 50, 50}, % \Theta
4715    "03 = {100,100}, % \Lambda
4716    "06 = { 50, 50}, % \Sigma
4717    "07 = {100,100}, % \Upsilon
4718    "08 = { 50, 50}, % \Phi
4719    "09 = { 50, 50}  % \Psi
```

Remaining slots can be found in the source file.

```
4720 ⟨/cmr⟩
4721    }
4722
4723 ⟨/m − t | cmr | pmn⟩
```

T1 and LY1 encodings contain some more characters. The default list will be loaded first.

```
4724 \SetProtrusion
4725 ⟨m − t⟩   [ name      = T1-default,
4726 ⟨bch⟩   [ name      = bch-T1,
4727 ⟨blg⟩   [ name      = blg-T1,
4728 ⟨cmr⟩   [ name      = cmr-T1,
4729 ⟨pad⟩   [ name      = pad-T1,
4730 ⟨pmn⟩   [ name      = pmnj-T1,
4731 ⟨ppl⟩   [ name      = ppl-T1,
4732 ⟨ptm⟩   [ name      = ptm-T1,
4733 ⟨ugm⟩   [ name      = ugm-T1,
4734 ⟨m − t⟩     load     = default    ]
```

```
4735 ⟨bch⟩      load      = bch-default ]
4736 ⟨blg⟩      load      = blg-default ]
4737 ⟨cmr⟩      load      = cmr-default ]
4738 ⟨pad⟩      load      = pad-default ]
4739 ⟨pmn⟩      load      = pmnj-default ]
4740 ⟨ppl⟩      load      = ppl-default ]
4741 ⟨ptm⟩      load      = ptm-default ]
4742 ⟨ugm⟩      load      = ugm-default ]
4743 ⟨m − t⟩    { encoding = {T1,LY1}    }
4744 ⟨bch | cmr | pad | pmn | ppl⟩    { encoding = {T1,LY1},
4745 ⟨blg | ptm | ugm⟩    { encoding = {T1},
4746 ⟨bch⟩      family    = bch }
4747 ⟨blg⟩      family    = blg }
4748 ⟨cmr⟩      family    = cmr }
4749 ⟨pad⟩      family    = {pad,padx,padj} }
4750 ⟨pmn⟩      family    = pmnj }
4751 ⟨ppl⟩      family    = {ppl,pplx,pplj} }
4752 ⟨ptm⟩      family    = {ptm,ptmx,ptmj} }
4753 ⟨ugm⟩      family    = ugm }
4754      {
4755 ⟨m − t | cmr⟩      \AE = {50,  },
4756 ⟨bch | pmn⟩      \OE = {50,  },
4757 ⟨pmn⟩      \TH = {  ,50},
4758 ⟨blg⟩      \v L = {   ,250},
4759 ⟨blg⟩      \v d = {   ,250},
4760 ⟨blg⟩      \v l = {   ,250},
4761 ⟨blg⟩      \v t = {   ,250},
4762 ⟨blg⟩      127 = {300,400},
4763 ⟨blg⟩      156 = {100,   }, % IJ
4764 ⟨blg⟩      188 = { 80, 80}, % ij
4765 ⟨m − t | bch | pad | pmn | ppl | ptm⟩      _ = {100,100},
4766 ⟨cmr⟩      _ = {200,200},
4767 ⟨ugm⟩      _ = {100,200},
4768 ⟨m − t | pad | pmn | ptm⟩      \textbackslash    = {100,200},
4769 ⟨bch⟩      \textbackslash    = {150,200},
4770 ⟨blg⟩      \textbackslash    = {250,300},
4771 ⟨cmr | ppl⟩      \textbackslash    = {200,300},
4772 ⟨ugm⟩      \textbackslash    = {100,300},
4773 ⟨ugm⟩      \textbar    = {200,200},
4774 ⟨blg⟩      \textendash    = {300,300},   \textemdash    = {150,150},
4775 ⟨blg⟩      \textquotedbl    = {300,400},   \textquotedblleft = {300,400},
4776 ⟨cmr⟩      \textquotedbl    = {300,300},   \textquotedblleft = {200,600},
```

The EC fonts do something weird: they insert an implicit kern between quote and
boundary character. Therefore, we must override the settings from OT1.

```
4777 ⟨m − t | cmr | pad | ppl | ptm | ugm⟩      \quotesinglbase   = {400,400},   \quotedblbase    = {400,400},
4778 ⟨blg⟩      \quotesinglbase   = {400,400},   \quotedblbase    = {300,400},
4779 ⟨bch | pmn⟩      \quotesinglbase   = {400,400},   \quotedblbase    = {300,300},
4780 ⟨m − t | bch | pmn⟩      \guilsinglleft    = {400,300},   \guilsinglright    = {300,400},
4781 ⟨blg⟩      \guilsinglleft    = {300,500},   \guilsinglright    = {300,500},
4782 ⟨cmr | pad | ppl | ptm⟩      \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
4783 ⟨ugm⟩      \guilsinglleft    = {400,400},   \guilsinglright    = {300,600},
4784 ⟨m − t⟩      \guillemotleft    = {200,200},   \guillemotright    = {200,200},
4785 ⟨cmr⟩      \guillemotleft    = {300,200},   \guillemotright    = {100,400},
4786 ⟨bch | pmn⟩      \guillemotleft    = {200,200},   \guillemotright    = {150,300},
4787 ⟨blg | pad | ppl | ptm⟩      \guillemotleft    = {300,300},   \guillemotright    = {200,400},
4788 ⟨ugm⟩      \guillemotleft    = {300,400},   \guillemotright    = {300,400},
4789 ⟨m − t | bch | cmr | pad | pmn | ppl | ugm⟩      \textexclamdown    = {100,   },   \textquestiondown = {100,   },
4790 ⟨blg⟩      \textexclamdown    = {200,   },   \textquestiondown = {100,   },
4791 ⟨ptm⟩      \textexclamdown    = {200,   },   \textquestiondown = {200,   },
```

```
4792 ⟨m − t | cmr | pad | ppl | ptm | ugm⟩      \textbraceleft     = {400,200},   \textbraceright     = {200,400},
4793 ⟨bch | blg | pmn⟩       \textbraceleft     = {200,  },   \textbraceright     = {   ,300},
4794 ⟨m − t | bch | cmr | pad | ppl | ptm | ugm⟩     \textless           = {200,100},   \textgreater        = {100,200}
4795 ⟨pmn⟩      \textless          = {100,   },   \textgreater        = {   ,100},
4796 ⟨pmn⟩      \textvisiblespace = {100,100} % not in LY1
4797    }
4798
```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```
4799 ⟨∗cmr⟩
4800 \SetProtrusion
4801    [ name        = lmr-T1,
4802      load        = cmr-T1    ]
4803    { encoding = {T1,LY1},
4804      family    = lmr       }
4805    {
4806      \textquotedblleft  = {300,400},  \textquotedblright = {300,400}
4807    }
4808
4809 ⟨/cmr⟩
```

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion).[20]

```
4810 ⟨∗m − t | cmr | pmn⟩
4811 \SetProtrusion
4812 ⟨m − t⟩   [ name      = T2A-default,
4813 ⟨cmr⟩    [ name      = cmr-T2A,
4814 ⟨pmn⟩    [ name      = pmnj-T2A,
4815 ⟨m − t⟩     load      = default     ]
4816 ⟨cmr⟩     load      = cmr-default ]
4817 ⟨pmn⟩     load      = pmnj-default ]
4818    { encoding = T2A,
4819 ⟨m − t⟩    }
4820 ⟨cmr⟩      family   = cmr }
4821 ⟨pmn⟩      family   = pmnj }
4822    {
4823      \CYRA = {50,50},
4824      \CYRG = {  ,50},
4825      \CYRK = {  ,50},
4826      \CYRT = {50,50},
4827      \CYRH = {50,50},
4828      \CYRU = {50,50},
4829 ⟨pmn⟩      \CYRS = {50,  },
4830 ⟨pmn⟩      \CYRO = {50,50},
4831      \cyrk = {  ,50},
4832      \cyrg = {  ,50},
4833      \cyrh = {50,50},
4834 ⟨m − t | pmn⟩      \cyru = {50,50},
4835 ⟨cmr⟩      \cyru = {50,70},
4836 ⟨m − t⟩         _  = {100,100},
4837 ⟨cmr⟩         _  = {200,200},
4838 ⟨m − t⟩      \textbackslash    = {100,200},   \quotedblbase      = {400,400},
4839 ⟨cmr⟩      \textbackslash    = {200,300},   \quotedblbase      = {400,400},
4840 ⟨pmn⟩      \textbackslash    = {100,200},   \quotedblbase      = {300,300},
4841 ⟨cmr⟩      \textquotedbl    = {300,300},   \textquotedblleft  = {200,600},
4842 ⟨m − t⟩      \guillemotleft    = {200,200},   \guillemotright    = {200,200},
```

---

20 Contributed by *Karl Karlsson.*

```
4843 ⟨cmr⟩      \guillemotleft     = {300,200},   \guillemotright    = {100,400},
4844 ⟨pmn⟩      \guillemotleft     = {200,200},   \guillemotright    = {150,300},
4845 ⟨m − t | cmr⟩     \textbraceleft     = {400,200},   \textbraceright    = {200,400},
4846 ⟨pmn⟩   \textbraceleft    = {200,  },   \textbraceright    = {   ,300},
4847 ⟨m − t | cmr⟩     \textless     = {200,100},   \textgreater       = {100,200}
4848 ⟨pmn⟩   \textless     = {100,  },   \textgreater       = {   ,100}
4849    }
4850
4851 ⟨/m − t | cmr | pmn⟩
```

Settings for the QX encoding (generic and Times).[21] It also includes some glyphs otherwise in TS1.

```
4852 ⟨∗m − t | ptm⟩
4853 \SetProtrusion
4854 ⟨m − t⟩   [ name      = QX-default,
4855 ⟨ptm⟩     [ name      = ptm-QX,
4856 ⟨m − t⟩      load      = default ]
4857 ⟨ptm⟩      load      = ptm-default ]
4858 ⟨m − t⟩   { encoding = QX }
4859 ⟨ptm⟩   { encoding = QX,
4860 ⟨ptm⟩       family    = {ptm,ptmx,ptmj} }
4861    {
4862       \AE = {50,  },
4863 ⟨ptm⟩      *   = {200,200},
4864       {=} = {100,100},
4865       \textunderscore    = {100,100},
4866       \textbackslash     = {100,200},
4867       \quotedblbase      = {400,400},
4868 ⟨m − t⟩      \guillemotleft     = {200,200},   \guillemotright    = {200,200},
4869 ⟨ptm⟩      \guillemotleft     = {300,300},   \guillemotright    = {200,400},
4870       \textexclamdown    = {100,  },   \textquestiondown  = {100,  },
4871 ⟨m − t⟩      \textbraceleft     = {400,200},   \textbraceright    = {200,400},
4872 ⟨ptm⟩      \textbraceleft     = {200,200},   \textbraceright    = {200,300},
4873       \textless     = {200,100},   \textgreater       = {100,200},
4874       \textminus    = {200,200},   \textdegree        = {300,300},
4875 ⟨m − t⟩      \copyright         = {100,100},   \textregistered    = {100,100}
4876 ⟨ptm⟩      \copyright    = {100,150},   \textregistered    = {100,150},
4877 ⟨ptm⟩      \textxgeq     = {   ,100},   \textxleq      = {100,   },
4878 ⟨ptm⟩      \textalpha    = {   , 50},   \textDelta     = { 70, 70},
4879 ⟨ptm⟩      \textpi       = { 50, 80},   \textSigma     = {   , 70},
4880 ⟨ptm⟩      \textmu       = {   , 80},   \texteuro      = { 50, 50},
4881 ⟨ptm⟩      \textellipsis     = {150,200},   \textasciitilde    = { 80, 80},
4882 ⟨ptm⟩      \textapprox   = { 50, 50},   \textinfty     = {100,100},
4883 ⟨ptm⟩      \textdagger   = {150,150},   \textdaggerdbl     = {100,100},
4884 ⟨ptm⟩      \textdiv      = { 50,150},   \textsection   = { 80, 80},
4885 ⟨ptm⟩      \texttimes    = {100,150},   \textpm        = { 50, 80},
4886 ⟨ptm⟩      \textbullet   = {150,150},   \textperiodcentered = {300,300},
4887 ⟨ptm⟩      \textquotesingle  = {500,500},   \textquotedbl      = {300,300},
4888 ⟨ptm⟩      \textperthousand  = {   ,50}
4889    }
4890
4891 ⟨/m − t | ptm⟩
```

T5 is based on OT1; it shares some but not all extra characters of T1. All accented characters are already taken care of by the inheritance list.

```
4892 ⟨∗cmr | bch⟩
4893 \SetProtrusion
4894 ⟨cmr⟩   [ name       = cmr-T5,
```

---

21   Contributed by *Maciej Eder.*

```
4895 ⟨cmr⟩      load     = cmr-default ]
4896 ⟨bch⟩   [ name     = bch-T5,
4897 ⟨bch⟩      load     = bch-default ]
4898    { encoding = T5,
4899 ⟨cmr⟩      family   = cmr }
4900 ⟨bch⟩      family   = bch }
4901    {
4902 ⟨bch⟩      _ = {100,100},
4903 ⟨bch⟩      \textbackslash    = {150,200},
4904 ⟨cmr⟩      \textbackslash    = {200,300},
4905 ⟨cmr⟩      \textquotedblleft = {200,600},
4906 ⟨cmr⟩      \textquotedbl     = {300,300},
4907 ⟨bch⟩      \quotesinglbase   = {400,400},   \quotedblbase        = {300,300},
4908 ⟨cmr⟩      \quotesinglbase   = {400,400},   \quotedblbase        = {400,400},
4909 ⟨bch⟩      \guilsinglleft    = {400,300},   \guilsinglright      = {300,400},
4910 ⟨cmr⟩      \guilsinglleft    = {400,400},   \guilsinglright      = {300,500},
4911 ⟨bch⟩      \guillemotleft    = {200,200},   \guillemotright      = {150,300},
4912 ⟨cmr⟩      \guillemotleft    = {300,200},   \guillemotright      = {100,400},
4913 ⟨bch⟩      \textbraceleft    = {200,  },    \textbraceright      = {   ,300},
4914 ⟨cmr⟩      \textbraceleft    = {400,200},   \textbraceright      = {200,400},
4915      \textless         = {200,100},   \textgreater         = {100,200}
4916    }
4917
4918 ⟨/cmr | bch⟩
```

Minion with lining numbers.

```
4919 ⟨∗pmn⟩
4920 \SetProtrusion
4921    [ name     = pmnx-OT1,
4922      load     = pmnj-default ]
4923    { encoding = OT1,
4924      family   = pmnx }
4925    {
4926      1 = {230,180}
4927    }
4928
4929 \SetProtrusion
4930    [ name     = pmnx-T1,
4931      load     = pmnj-T1  ]
4932    { encoding = {T1,LY1},
4933      family   = pmnx     }
4934    {
4935      1 = {230,180}
4936    }
4937
4938 \SetProtrusion
4939    [ name     = pmnx-T2A,
4940      load     = pmnj-T2A  ]
4941    { encoding = {T2A},
4942      family   = pmnx     }
4943    {
4944      1 = {230,180}
4945    }
4946
4947 ⟨/pmn⟩
```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```
4948 ⟨∗ptm⟩
4949  \SetProtrusion
```

```
4950    [ name      = ptm-LY1,
4951      load      = ptm-T1 ]
4952    { encoding = LY1,
4953      family    = {ptm,ptmx,ptmj} }
4954    {
4955      _                              = {100,100},
4956      \texttrademark               = {100,100},
4957      \textregistered              = {100,100},
4958      \textcopyright               = {100,100},
4959      \textdegree                  = {300,300},
4960      \textminus                   = {200,200},
4961      \textellipsis                = {150,200},
4962 %    \texteuro                    = {    ,    }, % ?
4963      \textcent                    = {100,100},
4964      \textquotesingle             = {500,500},
4965      \textflorin                  = { 50, 70},
4966      \textdagger                  = {150,150},
4967      \textdaggerdbl               = {100,100},
4968      \textperthousand             = {    , 50},
4969      \textbullet                  = {150,150},
4970      \textonesuperior             = {100,100},
4971      \texttwosuperior             = { 50, 50},
4972      \textthreesuperior           = { 50, 50},
4973      \textperiodcentered          = {300,300},
4974      \textplusminus               = { 50, 80},
4975      \textmultiply                = {100,100},
4976      \textdivide                  = { 50,150}
```

Remaining slots in the source file.

```
4977    }
4978
4979 ⟨/ptm⟩
```

15.8.2 Italics

To find default settings for italic is difficult, since the character shapes and their
behaviour at the beginning or end of line may be wildly different for different fonts.
Therefore, we leave the letters away, and only set up the punctuation characters.

```
4980 \SetProtrusion
4981 ⟨m − t⟩   [ name      = OT1-it   ]
4982 ⟨bch⟩     [ name      = bch-it   ]
4983 ⟨blg⟩     [ name      = blg-it,
4984 ⟨blg⟩       load      = blg-default ]
4985 ⟨cmr⟩     [ name      = cmr-it   ]
4986 ⟨pad⟩     [ name      = pad-it   ]
4987 ⟨pmn⟩     [ name      = pmnj-it  ]
4988 ⟨ppl⟩     [ name      = ppl-it   ]
4989 ⟨ptm⟩     [ name      = ptm-it   ]
4990 ⟨ugm⟩     [ name      = ugm-it   ]
4991 ⟨m − t | bch | blg | pad | ugm⟩   { encoding = OT1,
4992 ⟨ppl | ptm⟩   { encoding = {OT1,OT4},
4993 ⟨bch⟩        family    = bch,
4994 ⟨blg⟩        family    = blg,
4995 ⟨pad⟩        family    = {pad,padx,padj},
4996 ⟨ppl⟩        family    = {ppl,pplx,pplj},
4997 ⟨ptm⟩        family    = {ptm,ptmx,ptmj},
4998 ⟨ugm⟩        family    = ugm,
4999 ⟨m − t | bch | pad | ppl | ptm⟩      shape     = {it,sl} }
5000 ⟨blg | ugm⟩       shape     = it }
```

```
5001 ⟨cmr | pmn⟩    { }
5002      {
5003 ⟨cmr | ptm⟩       A = {100,50},
5004 ⟨pad | pmn⟩       A = {50,   },
5005 ⟨ugm⟩       A = {   ,150},
5006 ⟨ppl⟩       A = {50,50},
5007 ⟨ptm⟩    \AE = {100,   },
5008 ⟨pad | ppl⟩    \AE = {50,   },
5009 ⟨cmr | pad | ppl | ptm⟩       B = {50,   },
5010 ⟨pmn⟩       B = {20,-50},
5011 ⟨bch | ppl | ptm | ugm⟩       C = {50,   },
5012 ⟨cmr | pad⟩       C = {100, },
5013 ⟨pmn⟩       C = {50,-50},
5014 ⟨cmr | pad | ppl | ptm⟩       D = {50,50},
5015 ⟨pmn⟩       D = {20,   },
5016 ⟨cmr | pad | ppl | ptm⟩       E = {50,   },
5017 ⟨pmn⟩       E = {20,-50},
5018 ⟨cmr | pad | ptm⟩       F = {100, },
5019 ⟨pmn⟩       F = {10,   },
5020 ⟨ppl⟩       F = {50,   },
5021 ⟨bch | ppl | ptm | ugm⟩       G = {50,   },
5022 ⟨cmr | pad⟩       G = {100, },
5023 ⟨pmn⟩       G = {50,-50},
5024 ⟨cmr | pad | ppl | ptm⟩       H = {50,   },
5025 ⟨cmr | pad | ptm⟩       I = {50,   },
5026 ⟨pmn⟩       I = {20,-50},
5027 ⟨cmr | ptm⟩       J = {100, },
5028 ⟨pad⟩       J = {50,   },
5029 ⟨pmn⟩       J = {20,   },
5030 ⟨cmr | pad | ppl | ptm⟩       K = {50,   },
5031 ⟨pmn⟩       K = {20,   },
5032 ⟨cmr | pad | ppl | ptm⟩       L = {50,   },
5033 ⟨pmn⟩       L = {20,50},
5034 ⟨ugm⟩       L = {   ,100},
5035 ⟨cmr | ptm⟩       M = {50,   },
5036 ⟨pmn⟩       M = {   ,-30},
5037 ⟨cmr | ptm⟩       N = {50,   },
5038 ⟨pmn⟩       N = {   ,-30},
5039 ⟨bch | pmn | ppl | ptm⟩       O = {50,   },
5040 ⟨cmr | pad⟩       O = {100, },
5041 ⟨ugm⟩       O = {70,50},
5042 ⟨ppl | ptm⟩    \OE = {50,   },
5043 ⟨pad⟩    \OE = {100, },
5044 ⟨cmr | pad | ppl | ptm⟩       P = {50,   },
5045 ⟨pmn⟩       P = {20,-50},
5046 ⟨bch | pmn | ppl | ptm⟩       Q = {50,   },
5047 ⟨cmr | pad⟩       Q = {100, },
5048 ⟨ugm⟩       Q = {70,50},
5049 ⟨cmr | pad | ppl | ptm⟩       R = {50,   },
5050 ⟨pmn⟩       R = {20,   },
5051 ⟨bch | cmr | pad | ppl | ptm⟩       S = {50,   },
5052 ⟨pmn⟩       S = {20,-30},
5053 ⟨bch | cmr | pad | ppl | ptm⟩       $ = {50,   },
5054 ⟨pmn⟩       $ = {20,-30},
5055 ⟨bch | pmn | ugm⟩       T = {70,   },
5056 ⟨cmr | pad | ppl | ptm⟩       T = {100, },
5057 ⟨cmr | pad | ppl | ptm⟩       U = {50,   },
5058 ⟨pmn⟩       U = {50,-50},
5059 ⟨cmr | pad | pmn | ugm⟩       V = {100, },
5060 ⟨ppl | ptm⟩       V = {100,50},
```

```
5061 ⟨cmr | pad | pmn | ugm⟩      W = {100, },
5062 ⟨ppl⟩      W = {50,  },
5063 ⟨ptm⟩      W = {100,50},
5064 ⟨cmr | ppl | ptm⟩      X = {50,  },
5065 ⟨cmr | ptm⟩      Y = {100, },
5066 ⟨pmn⟩      Y = {50,  },
5067 ⟨ppl⟩      Y = {100,50},
5068 ⟨pmn⟩      Z = {  ,-50},
5069 ⟨pmn⟩      d = {  ,-50},
5070 ⟨pad | pmn⟩      f = { ,-100},
5071 ⟨pmn⟩      i = {  ,-30},
5072 ⟨pmn⟩      j = {  ,-30},
5073 ⟨pmn⟩      l = {  ,-100},
5074 ⟨bch⟩      o = {50,50},
5075 ⟨bch⟩      p = {  ,50},
5076 ⟨pmn⟩      p = {-50,  },
5077 ⟨bch⟩      q = {50,  },
5078 ⟨pmn⟩      r = {  ,50},
5079 ⟨bch⟩      t = {  ,50},
5080 ⟨pmn | ugm⟩      v = {50,  },
5081 ⟨bch⟩      w = {  ,50},
5082 ⟨pmn | ugm⟩      w = {50,  },
5083 ⟨bch⟩      y = {  ,50},
5084 ⟨cmr⟩      0 = {100, },
5085 ⟨bch | ptm⟩      1 = {150,100},
5086 ⟨cmr⟩      1 = {200,50},
5087 ⟨pad⟩      1 = {150, },
5088 ⟨pmn⟩      1 = {50,  },
5089 ⟨ppl⟩      1 = {100, },
5090 ⟨ugm⟩      1 = {150,150},
5091 ⟨cmr⟩      2 = {100,-100},
5092 ⟨pad | ppl | ptm⟩      2 = {50,  },
5093 ⟨pmn⟩      2 = {-50,  },
5094 ⟨bch⟩      3 = {50,  },
5095 ⟨cmr⟩      3 = {100,-100},
5096 ⟨pmn⟩      3 = {-100, },
5097 ⟨ptm⟩      3 = {100,50},
5098 ⟨bch⟩      4 = {100, },
5099 ⟨cmr | pad⟩      4 = {150, },
5100 ⟨ppl | ptm⟩      4 = {50,  },
5101 ⟨cmr⟩      5 = {100, },
5102 ⟨ptm⟩      5 = {50,  },
5103 ⟨bch⟩      6 = {50,  },
5104 ⟨cmr⟩      6 = {100, },
5105 ⟨bch | pad | ptm⟩      7 = {100, },
5106 ⟨cmr⟩      7 = {200,-150},
5107 ⟨pmn⟩      7 = {20,  },
5108 ⟨ppl⟩      7 = {50,  },
5109 ⟨cmr⟩      8 = {50,-50},
5110 ⟨cmr⟩      9 = {100,-100},
5111 ⟨m − t | cmr | pad | pmn | ppl⟩      . = { ,500},
5112 ⟨blg⟩      . = {400,600},
5113 ⟨bch | ptm | ugm⟩      . = { ,700},
5114 ⟨blg⟩      {,}= {300,500},
5115 ⟨m − t | cmr | pad | pmn | ppl⟩      {,}= { ,500},
5116 ⟨bch | ugm⟩      {,}= { ,600},
5117 ⟨ptm⟩      {,}= { ,700},
5118 ⟨m − t | cmr | pad | ppl⟩      : = { ,300},
5119 ⟨bch | ugm⟩      : = { ,400},
5120 ⟨pmn⟩      : = { ,200},
```

```
5121 ⟨ptm⟩        : = { ,500},
5122 ⟨m − t | cmr | pad | ppl⟩     ; = { ,300},
5123 ⟨bch | ugm⟩       ; = { ,400},
5124 ⟨pmn⟩       ; = { ,200},
5125 ⟨ptm⟩       ; = { ,500},
5126 ⟨ptm⟩       ! = { ,100},
5127 ⟨bch⟩       ? = { ,200},
5128 ⟨ptm⟩       ? = { ,100},
5129 ⟨ppl⟩       ? = { ,300},
5130 ⟨pmn⟩       " = {400,200},
5131 ⟨m − t | pad | pmn | ppl | ptm⟩     & = {50,50},
5132 ⟨bch⟩       & = {   ,80},
5133 ⟨cmr⟩       & = {100,50},
5134 ⟨ugm⟩       & = {50,100},
5135 ⟨m − t | cmr | pad | pmn⟩     \% = {100, },
5136 ⟨bch⟩       \% = {50,50},
5137 ⟨ppl | ptm⟩       \% = {100,100},
5138 ⟨ugm⟩       \% = {100,50},
5139 ⟨m − t | pmn | ppl⟩     * = {200,200},
5140 ⟨bch⟩       * = {300,200},
5141 ⟨cmr⟩       * = {400,100},
5142 ⟨pad⟩       * = {500,100},
5143 ⟨ptm | ugm⟩       * = {400,200},
5144 ⟨m − t | cmr | pmn | ppl⟩     + = {150,200},
5145 ⟨bch | ugm⟩       + = {250,250},
5146 ⟨pad | ptm⟩       + = {250,200},
5147 ⟨m − t | pad | pmn | ppl⟩     @ = {50,50},
5148 ⟨bch⟩       @ = {80,50},
5149 ⟨cmr⟩       @ = {200,50},
5150 ⟨ptm⟩       @ = {150,150},
5151 ⟨m − t | bch | ugm⟩     ~ = {150,150},
5152 ⟨cmr | pad | pmn | ppl | ptm⟩     ~ = {200,150},
5153 ⟨ugm⟩       {=}= {200,200},
5154 ⟨!blg⟩       ( = {200, },   ) = {   ,200},
5155 ⟨m − t | cmr | pad | ppl | ptm | ugm⟩     / = {100,200},
5156 ⟨bch⟩       / = {   ,150},
5157 ⟨pmn⟩       / = {100,150},
5158 ⟨m − t⟩       - = {300,300},
5159 ⟨bch | pad⟩       - = {300,400},
5160 ⟨pmn⟩       - = {200,300},
5161 ⟨cmr⟩       - = {500,300},
5162 ⟨ppl⟩       - = {300,500},
5163 ⟨ptm⟩       - = {500,500},
5164 ⟨ugm⟩       - = {400,700},
5165 ⟨blg⟩       _ = {0,300},
5166 ⟨m − t | pmn⟩     \textendash     = {200,200},   \textemdash     = {150,150},
5167 ⟨bch⟩     \textendash     = {200,300},   \textemdash     = {150,200},
5168 ⟨cmr⟩     \textendash     = {500,300},   \textemdash     = {400,200},
5169 ⟨pad | ppl | ptm | ugm⟩     \textendash     = {300,300},   \textemdash     = {200,200},
5170 ⟨m − t | bch | pmn | ugm⟩     \textquoteleft     = {400,200},   \textquoteright     = {400,200},
5171 ⟨blg⟩     \textquoteleft     = {400,400},   \textquoteright     = {400,400},
5172 ⟨cmr | pad⟩     \textquoteleft     = {800,200},   \textquoteright     = {800,200},
5173 ⟨ppl⟩     \textquoteleft     = {700,400},   \textquoteright     = {700,400},
5174 ⟨ptm⟩     \textquoteleft     = {800,500},   \textquoteright     = {800,500},
5175 ⟨m − t | bch | pmn⟩     \textquotedblleft = {400,200},   \textquotedblright = {400,200}
5176 ⟨blg⟩     \textquotedblright = {300,300}
5177 ⟨cmr⟩     \textquotedblleft = {700,100},   \textquotedblright = {500,300}
5178 ⟨pad⟩     \textquotedblleft = {700,200},   \textquotedblright = {700,200}
5179 ⟨ppl⟩     \textquotedblleft = {500,300},   \textquotedblright = {500,300}
5180 ⟨ptm⟩     \textquotedblleft = {700,400},   \textquotedblright = {700,400}
```

```
5181 ⟨ugm⟩      \textquotedblleft = {600,200},   \textquotedblright = {600,200}
5182    }
5183
5184 ⟨∗cmr | pmn⟩
5185 \SetProtrusion
5186 ⟨cmr⟩   [ name      = cmr-it-OT1,
5187 ⟨pmn⟩   [ name      = pmnj-it-OT1,
5188 ⟨cmr⟩     load      = cmr-it   ]
5189 ⟨pmn⟩     load      = pmnj-it  ]
5190 ⟨cmr⟩   { encoding = {OT1,OT4},
5191 ⟨pmn⟩    { encoding = OT1,
5192 ⟨cmr⟩     family   = cmr,
5193 ⟨pmn⟩     family   = pmnj,
5194 ⟨cmr⟩     shape    = it        }
5195 ⟨pmn⟩     shape    = {it,sl}  }
5196    {
5197 ⟨cmr⟩     \AE = {100,   },
5198 ⟨pmn⟩     \AE = {  ,-50},
5199 ⟨cmr⟩     \OE = {100,   },
5200 ⟨pmn⟩     \OE = {50,   }
5201 ⟨∗cmr⟩
5202      "00 = {200,150}, % \Gamma
5203      "01 = {150,100}, % \Delta
5204      "02 = {150, 50}, % \Theta
5205      "03 = {150, 50}, % \Lambda
5206      "04 = {100,100}, % \Xi
5207      "05 = {100,100}, % \Pi
5208      "06 = {100, 50}, % \Sigma
5209      "07 = {200,150}, % \Upsilon
5210      "08 = {150, 50}, % \Phi
5211      "09 = {150,100}, % \Psi
5212      "0A = { 50, 50}  % \Omega
5213 ⟨/cmr⟩
5214    }
5215
5216 ⟨/cmr | pmn⟩
5217 \SetProtrusion
5218 ⟨m − t⟩   [ name     = T1-it-default,
5219 ⟨bch⟩   [ name      = bch-it-T1,
5220 ⟨blg⟩   [ name      = blg-it-T1,
5221 ⟨cmr⟩   [ name      = cmr-it-T1,
5222 ⟨pad⟩   [ name      = pad-it-T1,
5223 ⟨pmn⟩   [ name      = pmnj-it-T1,
5224 ⟨ppl⟩   [ name      = ppl-it-T1,
5225 ⟨ptm⟩   [ name      = ptm-it-T1,
5226 ⟨ugm⟩   [ name      = ugm-it-T1,
5227 ⟨m − t⟩     load     = OT1-it  ]
5228 ⟨bch⟩     load      = bch-it   ]
5229 ⟨blg⟩     load      = blg-T1   ]
5230 ⟨cmr⟩     load      = cmr-it   ]
5231 ⟨pmn⟩     load      = pmnj-it  ]
5232 ⟨pad⟩     load      = pad-it   ]
5233 ⟨ppl⟩     load      = ppl-it   ]
5234 ⟨ptm⟩     load      = ptm-it   ]
5235 ⟨ugm⟩     load      = ugm-it   ]
5236 ⟨m − t | bch | cmr | pad | pmn | ppl⟩   { encoding = {T1,LY1},
5237 ⟨blg | ptm | ugm⟩   { encoding = T1,
5238 ⟨bch⟩     family   = bch,
5239 ⟨blg⟩     family   = blg,
5240 ⟨cmr⟩     family   = cmr,
```

```
5241 ⟨pmn⟩      family   = pmnj,
5242 ⟨pad⟩      family   = {pad,padx,padj},
5243 ⟨ppl⟩      family   = {ppl,pplx,pplj},
5244 ⟨ptm⟩      family   = {ptm,ptmx,ptmj},
5245 ⟨ugm⟩      family   = ugm,
5246 ⟨m − t | bch | pad | pmn | ppl | ptm⟩      shape    = {it,sl}  }
5247 ⟨blg | cmr | ugm⟩        shape   = it        }
5248    {
5249 ⟨m − t | bch | pmn⟩      _ = {  ,100},
5250 ⟨blg⟩       _ = {0,300},
5251 ⟨cmr | ugm⟩      _ = {100,200},
5252 ⟨pad | ppl | ptm⟩      _ = {100,100},
5253 ⟨blg⟩        . = {400,600},
5254 ⟨blg⟩      {,}= {300,500},
5255 ⟨cmr⟩        \AE = {100,   },
5256 ⟨pmn⟩        \AE = {  ,-50},
5257 ⟨bch | pmn⟩      \OE = { 50,   },
5258 ⟨cmr⟩        \OE = {100,   },
5259 ⟨pmn⟩        031 = { ,-100}, % ffl
5260 ⟨cmr | ptm⟩       156 = {100, },  % IJ
5261 ⟨pad⟩        156 = {50,  },  % IJ
5262 ⟨pmn⟩        156 = {20,  },  % IJ
5263 ⟨pmn⟩        188 = {  ,-30}, % ij
5264 ⟨pmn⟩      \v t = { ,100},
5265 ⟨m − t | pad | ppl | ptm⟩      \textbackslash    = {100,200},
5266 ⟨cmr | ugm⟩      \textbackslash    = {300,300},
5267 ⟨bch⟩      \textbackslash    = {150,150},
5268 ⟨pmn⟩      \textbackslash    = {100,150},
5269 ⟨ugm⟩      \textbar        = {200,200},
5270 ⟨cmr⟩      \textquotedblleft = {500,300},
5271 ⟨blg⟩      \textquoteleft    = {400,400},    \textquoteright    = {400,400},
5272 ⟨blg⟩      \textquotedbl     = {300,300},    \textquotedblleft  = {300,300},
5273 ⟨blg⟩      \textquotedblright = {300,300},    \quotedblbase      = {200,600},
5274 ⟨m − t | ptm⟩      \quotesinglbase  = {300,700},    \quotedblbase      = {400,500},
5275 ⟨cmr⟩      \quotesinglbase  = {300,700},    \quotedblbase      = {200,600},
5276 ⟨bch | pmn⟩      \quotesinglbase  = {200,500},    \quotedblbase      = {150,500},
5277 ⟨pad | ppl⟩      \quotesinglbase  = {500,500},    \quotedblbase      = {400,400},
5278 ⟨ugm⟩      \quotesinglbase  = {300,700},    \quotedblbase      = {300,500},
5279 ⟨m − t | ppl | ptm⟩      \guilsinglleft   = {400,400},    \guilsinglright    = {300,500},
5280 ⟨bch | pmn⟩      \guilsinglleft   = {300,400},    \guilsinglright    = {200,500},
5281 ⟨cmr⟩      \guilsinglleft   = {500,300},    \guilsinglright    = {400,400},
5282 ⟨pad⟩      \guilsinglleft   = {500,400},    \guilsinglright    = {300,500},
5283 ⟨ugm⟩      \guilsinglleft   = {400,400},    \guilsinglright    = {300,600},
5284 ⟨m − t | ppl⟩      \guillemotleft   = {300,300},    \guillemotright    = {300,300},
5285 ⟨bch | pmn⟩      \guillemotleft   = {200,300},    \guillemotright    = {150,400},
5286 ⟨cmr⟩      \guillemotleft   = {400,100},    \guillemotright    = {200,300},
5287 ⟨pad⟩      \guillemotleft   = {300,300},    \guillemotright    = {200,400},
5288 ⟨ptm⟩      \guillemotleft   = {300,400},    \guillemotright    = {200,400},
5289 ⟨ugm⟩      \guillemotleft   = {300,400},    \guillemotright    = {300,400},
5290 ⟨m − t | pad | ppl | ugm⟩      \textexclamdown  = {100,  },    \textquestiondown = {200,   },
5291 ⟨cmr | ptm⟩      \textexclamdown  = {200,  },    \textquestiondown = {200,   },
5292 ⟨pmn⟩      \textexclamdown  = {-50,  },    \textquestiondown = {-50,   },
5293 ⟨m − t | ppl | ugm⟩      \textbraceleft   = {200,100},    \textbraceright    = {200,200},
5294 ⟨bch | pmn⟩      \textbraceleft   = {200,  },    \textbraceright    = {  ,200},
5295 ⟨cmr | pad | ptm⟩      \textbraceleft   = {400,100},    \textbraceright    = {200,200},
5296 ⟨bch | pmn⟩      \textless        = {100,  },    \textgreater       = {  ,100},
5297 ⟨cmr | pad | ppl | ptm⟩      \textless        = {300,100},    \textgreater       = {200,100}
5298 ⟨pmn⟩      \textvisiblespace = {100,100}
5299    }
5300
```

```
5301 ⟨∗m − t | cmr | pmn⟩
5302 \SetProtrusion
5303 ⟨m − t⟩   [ name      = T2A-it-default,
5304 ⟨cmr⟩     [ name      = cmr-it-T2A,
5305 ⟨pmn⟩     [ name      = pmnj-it-T2A,
5306 ⟨m − t⟩      load      = OT1-it   ]
5307 ⟨cmr⟩       load      = cmr-it   ]
5308 ⟨pmn⟩       load      = pmnj-it  ]
5309    { encoding = T2A,
5310 ⟨cmr⟩       family   = cmr,
5311 ⟨pmn⟩       family   = pmnj,
5312 ⟨m − t | pmn⟩      shape    = {it,sl}  }
5313 ⟨cmr⟩       shape    = it        }
5314    {
5315 ⟨cmr⟩       \CYRA = {100,50},
5316 ⟨pmn⟩       \CYRA = {50,  },
5317 ⟨cmr⟩       \CYRB = {50,  },
5318 ⟨cmr⟩       \CYRV = {50,  },
5319 ⟨pmn⟩       \CYRV = {20,-50},
5320 ⟨cmr⟩       \CYRG = {100, },
5321 ⟨pmn⟩       \CYRG = {10,  },
5322 ⟨cmr⟩       \CYRD = {50,  },
5323 ⟨cmr⟩       \CYRE = {50,  },
5324 ⟨pmn⟩       \CYRE = {20,-50},
5325 ⟨cmr⟩       \CYRZH = {50,  },
5326 ⟨cmr⟩       \CYRZ = {50,  },
5327 ⟨pmn⟩       \CYRZ = {20,-50},
5328 ⟨cmr⟩       \CYRI = {50,  },
5329 ⟨pmn⟩       \CYRI = {  ,-30},
5330 ⟨cmr⟩       \CYRISHRT = {50,  },
5331 ⟨cmr⟩       \CYRK = {50,  },
5332 ⟨pmn⟩       \CYRK = {20,  },
5333 ⟨cmr⟩       \CYRL = {50,  },
5334 ⟨cmr⟩       \CYRM = {50,  },
5335 ⟨pmn⟩       \CYRM = {  ,-30},
5336 ⟨cmr⟩       \CYRN = {50,  },
5337 ⟨cmr⟩       \CYRO = {100, },
5338 ⟨pmn⟩       \CYRO = {50,  },
5339 ⟨cmr⟩       \CYRP = {50,  },
5340 ⟨cmr⟩       \CYRR = {50,  },
5341 ⟨pmn⟩       \CYRR = {20,-50},
5342 ⟨cmr⟩       \CYRS = {100, },
5343 ⟨pmn⟩       \CYRS = {50,  },
5344 ⟨cmr⟩       \CYRT = {100, },
5345 ⟨pmn⟩       \CYRT = {70,  },
5346 ⟨cmr⟩       \CYRU = {100, },
5347 ⟨pmn⟩       \CYRU = {50,  },
5348 ⟨cmr⟩       \CYRF = {100, },
5349 ⟨cmr⟩       \CYRH = {50,  },
5350 ⟨cmr⟩       \CYRC = {50,  },
5351 ⟨cmr⟩       \CYRCH = {100, },
5352 ⟨cmr⟩       \CYRSH = {50,  },
5353 ⟨cmr⟩       \CYRSHCH = {50,  },
5354 ⟨cmr⟩       \CYRHRDSN = {100, },
5355 ⟨cmr⟩       \CYRERY = {50,  },
5356 ⟨cmr⟩       \CYRSFTSN = {50,  },
5357 ⟨cmr⟩       \CYREREV = {50,  },
5358 ⟨cmr⟩       \CYRYU = {50,  },
5359 ⟨cmr⟩       \CYRYA = {50,  },
5360 ⟨pmn⟩       \CYRYA = {  ,20},
```

```
5361 ⟨pmn⟩      \cyrr = {-50,  },
5362 ⟨m − t | pmn⟩        _ = {  ,100},
5363 ⟨cmr⟩        _ = {100,200},
5364 ⟨pmn⟩        031 = { ,-100}, % ffl
5365 ⟨pmn⟩      \v t = {  ,100},
5366 ⟨m − t⟩      \textbackslash    = {100,200},   \quotedblbase      = {400,500},
5367 ⟨cmr⟩      \textbackslash    = {300,300},   \quotedblbase      = {200,600},
5368 ⟨pmn⟩      \textbackslash    = {100,150},   \quotedblbase      = {150,500},
5369 ⟨m − t⟩      \guillemotleft    = {300,300},   \guillemotright    = {300,300},
5370 ⟨cmr⟩      \guillemotleft    = {400,100},   \guillemotright    = {200,300},
5371 ⟨pmn⟩      \guillemotleft    = {200,300},   \guillemotright    = {150,400},
5372 ⟨m − t⟩      \textbraceleft    = {200,100},   \textbraceright    = {200,200},
5373 ⟨cmr⟩      \textbraceleft    = {400,100},   \textbraceright    = {200,200},
5374 ⟨pmn⟩      \textbraceleft    = {200,  },   \textbraceright    = {   ,200},
5375 ⟨cmr⟩      \textquotedblleft = {500,300},
5376 ⟨cmr⟩      \textless         = {300,100},   \textgreater       = {200,100}
5377 ⟨pmn⟩      \textless         = {100,  },   \textgreater       = {   ,100}
5378   }
5379
5380 ⟨/m − t | cmr | pmn⟩
5381 ⟨∗m − t | ptm⟩
5382 \SetProtrusion
5383 ⟨m − t⟩   [ name     = QX-it-default,
5384 ⟨ptm⟩   [ name     = ptm-it-QX,
5385 ⟨m − t⟩       load    = OT1-it ]
5386 ⟨ptm⟩       load    = ptm-it ]
5387   { encoding = {QX},
5388 ⟨ptm⟩       family   = {ptm,ptmx,ptmj},
5389     shape    = {it,sl} }
5390   {
5391 ⟨ptm⟩      009 = {  , 50}, % fk
5392     {=} = {100,100},
5393 ⟨m − t⟩      \textunderscore   = {100,100},
5394 ⟨ptm⟩      \textunderscore   = {100,150},
5395     \textbackslash    = {100,200},
5396     \quotedblbase     = {300,400},
5397 ⟨m − t⟩      \guillemotleft    = {300,300},   \guillemotright    = {300,300},
5398 ⟨ptm⟩      \guillemotleft    = {200,400},   \guillemotright    = {200,400},
5399     \textexclamdown   = {200,  },   \textquestiondown  = {200,  },
5400     \textbraceleft    = {200,100},   \textbraceright    = {200,200},
5401     \textless         = {100,100},   \textgreater       = {100,100},
5402     \textminus        = {200,200},   \textdegree        = {300,150},
5403 ⟨m − t⟩      \copyright        = {100,100},   \textregistered    = {100,100}
5404 ⟨ptm⟩      \textregistered   = {100,150},   \copyright         = {100,150},
5405 ⟨ptm⟩      \textDelta        = { 70,  },   \textdelta         = {   , 50},
5406 ⟨ptm⟩      \textpi           = { 50, 80},   \textmu            = {   , 80},
5407 ⟨ptm⟩      \texteuro         = {200,  },   \textellipsis      = {100,200},
5408 ⟨ptm⟩      \textquoteleft    = {500,400},   \textquoteright    = {500,400},
5409 ⟨ptm⟩      \textquotedblleft = {500,300},   \textquotedblright = {400,400},
5410 ⟨ptm⟩      \textapprox       = { 50, 50},   \textinfty         = {100,100},
5411 ⟨ptm⟩      \textdagger       = {150,150},   \textdaggerdbl     = {100,100},
5412 ⟨ptm⟩      \textdiv          = {150,150},   \textasciitilde    = { 80, 80},
5413 ⟨ptm⟩      \texttimes        = {100,150},   \textpm            = { 50, 80},
5414 ⟨ptm⟩      \textbullet       = {300,100},   \textperiodcentered = {300,300},
5415 ⟨ptm⟩      \textquotesingle  = {500,500},   \textquotedbl      = {300,300},
5416 ⟨ptm⟩      \textperthousand  = {   ,50}
5417   }
5418
5419 ⟨/m − t | ptm⟩
5420 ⟨∗cmr | bch⟩
```

```
5421 \SetProtrusion
5422 ⟨cmr⟩    [ name = cmr-it-T5,
5423 ⟨cmr⟩      load = cmr-it ]
5424 ⟨bch⟩    [ name = bch-it-T5,
5425 ⟨bch⟩      load = bch-it ]
5426    { encoding = T5,
5427 ⟨bch⟩      family   = bch,
5428 ⟨cmr⟩      family   = cmr,
5429      shape    = it }
5430    {
5431 ⟨bch⟩        _ = {   ,100},
5432 ⟨cmr⟩        _ = {100,200},
5433 ⟨bch⟩    \textbackslash    = {150,150},
5434 ⟨cmr⟩    \textbackslash    = {300,300},
5435 ⟨bch⟩    \quotesinglbase   = {200,500},    \quotedblbase      = {150,500},
5436 ⟨cmr⟩    \quotesinglbase   = {300,700},    \quotedblbase      = {200,600},
5437 ⟨bch⟩    \guilsinglleft    = {300,400},    \guilsinglright    = {200,500},
5438 ⟨cmr⟩    \guilsinglleft    = {500,300},    \guilsinglright    = {400,400},
5439 ⟨bch⟩    \guillemotleft    = {200,300},    \guillemotright    = {150,400},
5440 ⟨cmr⟩    \guillemotleft    = {400,100},    \guillemotright    = {200,300},
5441 ⟨bch⟩    \textbraceleft    = {200, },      \textbraceright    = {   ,200},
5442 ⟨cmr⟩    \textbraceleft    = {400,100},    \textbraceright    = {200,200},
5443 ⟨bch⟩    \textless         = {100, },      \textgreater       = {   ,100}
5444 ⟨cmr⟩    \textless         = {300,100},    \textgreater       = {200,100}
5445    }
5446
5447 ⟨/cmr | bch⟩
```

Slanted is very similar to italic.

```
5448 ⟨∗cmr⟩
5449 \SetProtrusion
5450    [ name      = cmr-sl,
5451      load      = cmr-it-OT1 ]
5452    { encoding = {OT1,OT4},
5453      family   = cmr,
5454      shape    = sl  }
5455    {
5456      L = {  ,50},
5457      f = { ,-50},
5458      - = {300, },
5459      \textendash = {400,  },  \textemdash = {300,  }
5460    }
5461
5462 \SetProtrusion
5463    [ name      = cmr-sl-T1,
5464      load      = cmr-it-T1 ]
5465    { encoding = {T1,LY1},
5466      family   = cmr,
5467      shape    = sl  }
5468    {
5469      L = {  ,50},
5470      f = { ,-50},
5471      - = {300, },
5472      \textendash = {400,  },  \textemdash = {300,  }
5473    }
5474
5475 \SetProtrusion
5476    [ name      = cmr-sl-T2A,
5477      load      = cmr-it-T2A ]
5478    { encoding = T2A,
```

```
5479       family    = cmr,
5480       shape     = sl  }
5481     {
5482       L = {  ,50},
5483       f = { ,-50},
5484       - = {300, },
5485       \textendash = {400,  },  \textemdash = {300,  }
5486     }
5487
5488  \SetProtrusion
5489     [ name      = cmr-sl-T5,
5490       load      = cmr-it-T5 ]
5491     { encoding = T5,
5492       family    = cmr,
5493       shape     = sl  }
5494     {
5495       L = {  ,50},
5496       f = { ,-50},
5497       - = {300, },
5498       \textendash = {400,  },  \textemdash = {300,  }
5499     }
5500
5501  \SetProtrusion
5502     [ name      = lmr-it-T1,
5503       load      = cmr-it-T1 ]
5504     { encoding = {T1,LY1},
5505       family    = lmr,
5506       shape     = {it,sl} }
5507     {
5508       \textquotedblleft  = { ,200},  \textquotedblright = { ,200},
5509       \quotesinglbase    = { ,400},  \quotedblbase       = { ,500}
5510     }
5511
```

Oldstyle numerals are slightly different.

```
5512  \SetProtrusion
5513     [ name = cmr(oldstyle)-it,
5514       load = cmr-it-T1 ]
5515     { encoding = T1,
5516       family    = {hfor,cmor},
5517       shape     = {it,sl}  }
5518     {
5519       1 = {250, 50},
5520       2 = {150,-100},
5521       3 = {100,-50},
5522       4 = {150,150},
5523       6 = {200,    },
5524       7 = {200, 50},
5525       8 = {150,-50},
5526       9 = {100, 50}
5527     }
5528
5529  ⟨/cmr⟩
5530  ⟨∗pmn⟩
5531  \SetProtrusion
5532     [ name      = pmnx-it,
5533       load      = pmnj-it ]
5534     { encoding = OT1,
5535       family    = pmnx,
5536       shape     = {it,sl} }
```

```
5537     {
5538       1 = {100,150}
5539     }
5540
5541 \SetProtrusion
5542     [ name      = pmnx-it-T1,
5543       load      = pmnj-it-T1 ]
5544     { encoding = {T1,LY1},
5545       family    = pmnx,
5546       shape     = {it,sl} }
5547     {
5548       1 = {100,150}
5549     }
5550
5551 \SetProtrusion
5552     [ name      = pmnx-it-T2A,
5553       load      = pmnj-it-T2A ]
5554     { encoding = {T2A},
5555       family    = pmnx,
5556       shape     = {it,sl} }
5557     {
5558       1 = {100,150}
5559     }
5560
5561 ⟨/pmn⟩
5562 ⟨*ptm⟩
5563 \SetProtrusion
5564     [ name      = ptm-it-LY1,
5565       load      = ptm-it-T1    ]
5566     { encoding = {LY1},
5567       family    = {ptm,ptmx,ptmj},
5568       shape     = {it,sl}  }
5569     {
5570       _                        = {100,100},
5571       \texttrademark           = {100,100},
5572       \textregistered          = {100,100},
5573       \textcopyright           = {100,100},
5574       \textdegree              = {300,100},
5575       \textminus               = {200,200},
5576       \textellipsis            = {100,200},
5577 %     \texteuro                = {    ,    }, % ?
5578       \textcent                = {100,100},
5579       \textquotesingle         = {500,    },
5580       \textflorin              = {100, 70},
5581       \textdagger              = {150,150},
5582       \textdaggerdbl           = {100,100},
5583       \textbullet              = {150,150},
5584       \textonesuperior         = {150,100},
5585       \texttwosuperior         = {150, 50},
5586       \textthreesuperior       = {150, 50},
5587       \textparagraph           = {100,    },
5588       \textperiodcentered      = {500,300},
5589       \textonequarter          = { 50,    },
5590       \textonehalf             = { 50,    },
5591       \textplusminus           = {100,100},
5592       \textmultiply            = {150,150},
5593       \textdivide              = {150,150}
5594     }
5595
5596 ⟨/ptm⟩
```

15.8.3   Small caps

Small caps should inherit the values from their big brothers. Since values are relative to character width, we don't need to adjust them any further (but we have to reset some characters).

```
5597  ⟨∗!(blg | ugm)⟩
5598  \SetProtrusion
5599  ⟨m − t⟩    [ name     = OT1-sc,
5600  ⟨bch⟩    [ name     = bch-sc,
5601  ⟨cmr⟩    [ name     = cmr-sc-OT1,
5602  ⟨pad⟩    [ name     = pad-sc,
5603  ⟨pmn⟩     [ name     = pmnj-sc,
5604  ⟨ppl⟩    [ name     = ppl-sc,
5605  ⟨ptm⟩    [ name     = ptm-sc,
5606  ⟨m − t⟩     load      = default ]
5607  ⟨bch⟩    load      = bch-default ]
5608  ⟨cmr⟩    load      = cmr-OT1 ]
5609  ⟨pad⟩    load      = pad-default ]
5610  ⟨pmn⟩     load      = pmnj-default ]
5611  ⟨ppl⟩     load      = ppl-default ]
5612  ⟨ptm⟩     load      = ptm-default ]
5613  ⟨m − t | bch | pad | pmn⟩    { encoding = OT1,
5614  ⟨cmr | ppl | ptm⟩     { encoding = {OT1,OT4},
5615  ⟨bch⟩     family    = bch,
5616  ⟨cmr⟩     family    = cmr,
5617  ⟨pad⟩     family    = {pad,padx,padj},
5618  ⟨pmn⟩      family    = pmnj,
5619  ⟨ppl⟩     family    = {ppl,pplx,pplj},
5620  ⟨ptm⟩      family    = {ptm,ptmx,ptmj},
5621       shape     = sc }
5622     {
5623       a = {50,50},
5624  ⟨cmr | pad | ppl | ptm⟩    \ae = {50,  },
5625  ⟨bch | pmn⟩     c = {50,  },
5626  ⟨bch | pad | pmn⟩      d = {  ,50},
5627  ⟨m − t | bch | cmr | pad | pmn | ptm⟩      f = {  ,50},
5628  ⟨bch | pad | pmn⟩      g = {50,  },
5629  ⟨m − t | cmr | pad | pmn | ppl | ptm⟩     j = {50,  },
5630  ⟨bch⟩     j = {100,  },
5631  ⟨m − t | bch | cmr | pad | pmn | ppl⟩     l = {  ,50},
5632  ⟨ptm⟩     l = {  ,80},
5633  ⟨m − t | bch | cmr | pad | pmn | ppl⟩    013 = {  ,50}, % fl
5634  ⟨ptm⟩    013 = {  ,80}, % fl
5635  ⟨bch | pad | pmn⟩      o = {50,50},
5636  ⟨pad | pmn⟩     \oe = {50,  },
5637  ⟨ppl⟩      p = { 0, 0},
5638  ⟨bch | pad | pmn⟩      q = {50,70},
5639  ⟨ppl⟩      q = { 0,  },
5640  ⟨m − t | cmr | pad | pmn | ppl | ptm⟩     r = {  , 0},
5641       t = {50,50},
5642  ⟨m − t | bch | cmr | pad | pmn | ppl⟩     y = {50,50}
5643  ⟨ptm⟩     y = {80,80}
5644     }
5645
5646  \SetProtrusion
5647  ⟨m − t⟩    [ name     = T1-sc,
5648  ⟨bch⟩    [ name     = bch-sc-T1,
5649  ⟨cmr⟩    [ name     = cmr-sc-T1,
5650  ⟨pad⟩    [ name     = pad-sc-T1,
```

```
5651 ⟨pmn⟩   [ name      = pmnj-sc-T1,
5652 ⟨ppl⟩   [ name      = ppl-sc-T1,
5653 ⟨ptm⟩   [ name      = ptm-sc-T1,
5654 ⟨m − t⟩     load      = T1-default ]
5655 ⟨bch⟩     load     = bch-T1     ]
5656 ⟨cmr⟩     load     = cmr-T1     ]
5657 ⟨pad⟩     load     = pad-T1     ]
5658 ⟨pmn⟩      load     = pmnj-T1    ]
5659 ⟨ppl⟩      load     = ppl-T1     ]
5660 ⟨ptm⟩      load     = ptm-T1     ]
5661    { encoding = {T1,LY1},
5662 ⟨bch⟩     family   = bch,
5663 ⟨cmr⟩     family   = cmr,
5664 ⟨pad⟩     family   = {pad,padx,padj},
5665 ⟨pmn⟩      family   = pmnj,
5666 ⟨ppl⟩      family   = {ppl,pplx,pplj},
5667 ⟨ptm⟩      family   = {ptm,ptmx,ptmj},
5668      shape    = sc }
5669    {
5670      a = {50,50},
5671 ⟨cmr | pad | ppl | ptm⟩   \ae = {50,  },
5672 ⟨bch | pmn⟩        c = {50,  },
5673 ⟨bch | pad | pmn⟩       d = {  ,50},
5674 ⟨m − t | bch | cmr | pad | pmn | ptm⟩      f = {  ,50},
5675 ⟨bch | pad | pmn⟩       g = {50,  },
5676 ⟨m − t | cmr | pad | pmn | ppl | ptm⟩      j = {50,  },
5677 ⟨bch⟩       j = {100,  },
5678 ⟨m − t | bch | cmr | pad | pmn | ppl⟩      l = {  ,50},
5679 ⟨ptm⟩       l = {  ,80},
5680 ⟨m − t | bch | cmr | pad | pmn | ppl⟩  029 = {  ,50}, % fl
5681 ⟨ptm⟩     029 = {  ,80}, % fl
5682 ⟨bch | pad | pmn⟩       o = {50,50},
5683 ⟨bch | pad | pmn⟩       \oe = {50,  },
5684 ⟨ppl⟩       p = { 0, 0},
5685 ⟨bch | pad | pmn⟩       q = {50,70},
5686 ⟨ppl⟩       q = { 0,  },
5687 ⟨m − t | cmr | pad | pmn | ppl | ptm⟩      r = {  , 0},
5688      t = {50,50},
5689 ⟨m − t | bch | cmr | pad | pmn | ppl⟩      y = {50,50}
5690 ⟨ptm⟩     y = {80,80}
5691    }
5692
5693 ⟨/!(blg | ugm)⟩
5694 ⟨∗m − t | cmr⟩
5695 \SetProtrusion
5696 ⟨m − t⟩   [ name      = T2A-sc,
5697 ⟨cmr⟩   [ name      = cmr-sc-T2A,
5698 ⟨m − t⟩     load      = T2A-default ]
5699 ⟨cmr⟩     load     = cmr-T2A      ]
5700    { encoding = T2A,
5701 ⟨cmr⟩     family   = cmr,
5702      shape    = sc }
5703    {
5704      \cyra = {50,50},
5705      \cyrg = {  ,50},
5706      \cyrt = {50,50},
5707      \cyry = {  ,50}
5708    }
5709
5710 ⟨/m − t | cmr⟩
```

```
5711 ⟨∗m − t⟩
5712 \SetProtrusion
5713    [ name      = QX-sc,
5714      load      = QX-default ]
5715    { encoding = QX,
5716      shape     = sc  }
5717    {
5718      a = {50,50},
5719      f = {  ,50},
5720      j = {50,  },
5721      l = {  ,50},
5722    013 = {  ,50}, % fl
5723      r = {  , 0},
5724      t = {50,50},
5725      y = {50,50}
5726    }
5727
5728 ⟨/m − t⟩
5729 ⟨∗cmr | bch⟩
5730 \SetProtrusion
5731 ⟨bch⟩    [ name      = bch-sc-T5,
5732 ⟨bch⟩      load      = bch-T5 ]
5733 ⟨cmr⟩    [ name      = cmr-sc-T5,
5734 ⟨cmr⟩      load      = cmr-T5 ]
5735    { encoding = T5,
5736 ⟨bch⟩      family   = bch,
5737 ⟨cmr⟩      family   = cmr,
5738      shape     = sc }
5739    {
5740      a = {50,50},
5741 ⟨bch⟩      c = {50,  },
5742 ⟨bch⟩      d = {  ,50},
5743      f = {  ,50},
5744 ⟨bch⟩      g = {50,  },
5745 ⟨bch⟩      j = {100,  },
5746 ⟨cmr⟩      j = {50,  },
5747      l = {  ,50},
5748 ⟨bch⟩      o = {50,50},
5749 ⟨bch⟩      q = { 0,  },
5750 ⟨cmr⟩      r = {  , 0},
5751      t = {50,50},
5752      y = {50,50}
5753    }
5754
5755 ⟨/cmr | bch⟩
5756 ⟨∗pmn⟩
5757 \SetProtrusion
5758    [ name      = pmnx-sc,
5759      load      = pmnj-sc ]
5760    { encoding = OT1,
5761      family   = pmnx,
5762      shape     = sc }
5763    {
5764      1 = {230,180}
5765    }
5766
5767 \SetProtrusion
5768    [ name      = pmnx-sc-T1,
5769      load      = pmnj-sc-T1 ]
5770    { encoding = {T1,LY1},
```

```
5771       family   = pmnx,
5772       shape    = sc }
5773    {
5774       1 = {230,180}
5775    }
5776
```

15.8.4 Italic small caps

Minion provides real small caps in italics. The slantsc package calls them scit,
Philipp Lehman's fontinstallationguide suggests si.

```
5777 \SetProtrusion
5778    [ name     = pmnj-scit,
5779      load     = pmnj-it   ]
5780    { encoding = OT1,
5781      family   = pmnj,
5782      shape    = {scit,si} }
5783    {
5784      a = {50,  },
5785    \ae = {  ,-50},
5786      b = {20,-50},
5787      c = {50,-50},
5788      d = {20, 0},
5789      e = {20,-50},
5790      f = {10, 0},
5791    012 = {10,-50}, % fi
5792    013 = {10,-50}, % fl
5793    014 = {10,-50}, % ffi
5794    015 = {10,-50}, % ffl
5795      g = {50,-50},
5796      i = {20,-50},
5797      j = {20, 0},
5798      k = {20,  },
5799      l = {20,50},
5800      m = {  ,-30},
5801      n = {  ,-30},
5802      o = {50,  },
5803    \oe = {50,-50},
5804      p = {20,-50},
5805      q = {50,  },
5806      r = {20, 0},
5807      s = {20,-30},
5808      t = {70,  },
5809      u = {50,-50},
5810      v = {100,  },
5811      w = {100,  },
5812      y = {50,  },
5813      z = {  ,-50}
5814    }
5815
5816 \SetProtrusion
5817    [ name     = pmnj-scit-T1,
5818      load     = pmnj-it-T1   ]
5819    { encoding = {T1,LY1},
5820      family   = pmnj,
5821      shape    = {scit,si}    }
5822    {
5823      a = {50,  },
5824    \ae = {  ,-50},
```

```
5825        b = {20,-50},
5826        c = {50,-50},
5827        d = {20, 0},
5828        e = {20,-50},
5829        f = {10, 0},
5830      028 = {10,-50}, % fi
5831      029 = {10,-50}, % fl
5832      030 = {10,-50}, % ffi
5833      031 = {10,-50}, % ffl
5834        g = {50,-50},
5835        i = {20,-50},
5836      188 = {20, 0},  % ij
5837        j = {20, 0},
5838        k = {20,  },
5839        l = {20,50},
5840        m = {  ,-30},
5841        n = {  ,-30},
5842        o = {50,  },
5843      \oe = {50,-50},
5844        p = {20,-50},
5845        q = {50,  },
5846        r = {20, 0},
5847        s = {20,-30},
5848        t = {70,  },
5849        u = {50,-50},
5850        v = {100,  },
5851        w = {100,  },
5852        y = {50,  },
5853        z = {  ,-50}
5854      }
5855
5856  \SetProtrusion
5857     [ name      = pmnx-scit,
5858       load      = pmnj-scit ]
5859     { encoding = OT1,
5860       family   = pmnx,
5861       shape    = {scit,si} }
5862     {
5863       1 = {100,150}
5864     }
5865
5866  \SetProtrusion
5867     [ name      = pmnx-scit-T1,
5868       load      = pmnj-scit-T1 ]
5869     { encoding = {T1,LY1},
5870       family   = pmnx,
5871       shape    = {scit,si}    }
5872     {
5873       1 = {100,150}
5874     }
5875
5876  ⟨/pmn⟩
```

15.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```
5877  \SetProtrusion
5878  ⟨m − t⟩   [ name      = textcomp ]
```

```
5879 ⟨bch⟩    [ name      = bch-textcomp ]
5880 ⟨blg⟩    [ name      = blg-textcomp ]
5881 ⟨cmr⟩    [ name      = cmr-textcomp ]
5882 ⟨pad⟩    [ name      = pad-textcomp ]
5883 ⟨pmn⟩    [ name      = pmn-textcomp ]
5884 ⟨ppl⟩    [ name      = ppl-textcomp ]
5885 ⟨ptm⟩    [ name      = ptm-textcomp ]
5886 ⟨ugm⟩    [ name      = ugm-textcomp ]
5887 ⟨m − t⟩   { encoding = TS1       }
5888 ⟨!m − t⟩  { encoding = TS1,
5889 ⟨bch⟩       family   = bch }
5890 ⟨blg⟩       family   = blg }
5891 ⟨cmr⟩       family   = cmr }
5892 ⟨pad⟩       family   = {pad,padx,padj} }
5893 ⟨pmn⟩       family   = {pmnx,pmnj} }
5894 ⟨ppl⟩       family   = {ppl,pplx,pplj} }
5895 ⟨ptm⟩       family   = {ptm,ptmx,ptmj} }
5896 ⟨ugm⟩       family   = ugm }
5897    {
5898 ⟨blg⟩       \textquotestraightbase    = {400,500},
5899 ⟨cmr⟩       \textquotestraightbase    = {300,300},
5900 ⟨pad | pmn⟩     \textquotestraightbase    = {400,400},
5901 ⟨blg⟩       \textquotestraightdblbase = {300,400},
5902 ⟨cmr | pmn⟩     \textquotestraightdblbase = {300,300},
5903 ⟨pad⟩       \textquotestraightdblbase = {400,400},
5904 ⟨bch | cmr | pad | pmn | ugm⟩     \texttwelveudash          = {200,200},
5905 ⟨bch | cmr | pad | pmn⟩       \textthreequartersemdash = {150,150},
5906 ⟨ugm⟩       \textthreequartersemdash = {200,200},
5907 ⟨blg⟩       \textquotesingle          = {500,600},
5908 ⟨cmr | pmn⟩     \textquotesingle          = {300,400},
5909 ⟨pad⟩       \textquotesingle          = {400,500},
5910 ⟨ptm⟩       \textquotesingle          = {500,500},
5911 ⟨ugm⟩       \textquotesingle          = {300,500},
5912 ⟨bch | cmr | pmn⟩     \textasteriskcentered     = {200,300},
5913 ⟨blg⟩       \textasteriskcentered     = {150,200},
5914 ⟨pad⟩       \textasteriskcentered     = {300,300},
5915 ⟨ugm⟩       \textasteriskcentered     = {100,200},
5916 ⟨pmn⟩       \textfractionsolidus      = {-200,-200},
5917 ⟨cmr⟩       \textoneoldstyle          = {100,100},
5918 ⟨pmn⟩       \textoneoldstyle          = {   , 50},
5919 ⟨cmr⟩       \textthreeoldstyle        = {   , 50},
5920 ⟨pad | pmn⟩     \textthreeoldstyle        = { 50,   },
5921 ⟨cmr⟩       \textfouroldstyle         = { 50, 50},
5922 ⟨pad | pmn⟩     \textfouroldstyle         = { 50,   },
5923 ⟨cmr | pad | pmn⟩     \textsevenoldstyle        = { 50, 80},
5924 ⟨cmr⟩       \textlangle               = {400,   },
5925 ⟨cmr⟩       \textrangle               = {   ,400},
5926 ⟨m − t | bch | pmn | ptm⟩     \textminus                = {200,200},
5927 ⟨cmr | pad | ppl⟩     \textminus                = {300,300},
5928 ⟨blg | ugm⟩     \textminus                = {250,300},
5929 ⟨bch | pad | pmn⟩     \textlbrackdbl            = {100,   },
5930 ⟨blg⟩       \textlbrackdbl            = {200,   },
5931 ⟨bch | pad | pmn⟩     \textrbrackdbl            = {   ,100},
5932 ⟨blg⟩       \textrbrackdbl            = {   ,200},
5933 ⟨pmn⟩       \textasciigrave           = {200,500},
5934 ⟨bch | blg | cmr | pad | pmn⟩     \texttildelow             = {200,250},
5935 ⟨pmn⟩       \textasciibreve           = {300,400},
5936 ⟨pmn⟩       \textasciicaron           = {300,400},
5937 ⟨pmn⟩       \textacutedbl             = {200,300},
5938 ⟨pmn⟩       \textgravedbl             = {150,300},
```

```
5939 ⟨bch | pmn | ugm⟩      \textdagger                = {  80,  80},
5940 ⟨blg⟩       \textdagger           = {200,200},
5941 ⟨cmr | pad⟩      \textdagger                = {100,100},
5942 ⟨ptm⟩      \textdagger            = {150,150},
5943 ⟨blg⟩      \textdaggerdbl        = {150,150},
5944 ⟨cmr | pad | pmn⟩      \textdaggerdbl          = {  80,  80},
5945 ⟨ptm⟩      \textdaggerdbl        = {100,100},
5946 ⟨bch⟩      \textbardbl           = {100,100},
5947 ⟨blg | ugm⟩      \textbardbl             = {150,150},
5948 ⟨bch⟩      \textbullet           = {200,200},
5949 ⟨blg⟩      \textbullet           = {400,500},
5950 ⟨cmr | pad | pmn⟩      \textbullet             = {    ,100},
5951 ⟨ptm⟩      \textbullet           = {150,150},
5952 ⟨ugm⟩      \textbullet           = {  50,100},
5953 ⟨bch | cmr | pmn⟩      \textcelsius            = {  50,    },
5954 ⟨pad⟩      \textcelsius          = {  80,    },
5955 ⟨bch⟩      \textflorin           = {  50,  50},
5956 ⟨blg⟩      \textflorin           = {100,100},
5957 ⟨pad | ugm⟩      \textflorin             = {    ,100},
5958 ⟨pmn⟩      \textflorin           = {  50,100},
5959 ⟨ptm⟩      \textflorin           = {  50,  70},
5960 ⟨cmr⟩      \textcolonmonetary    = {    ,  50},
5961 ⟨pad | pmn⟩      \textcolonmonetary      = {  50,    },
5962 ⟨pmn⟩      \textinterrobang      = {    ,100},
5963 ⟨pmn⟩      \textinterrobangdown  = {100,    },
5964 ⟨m − t | pad | ptm⟩      \texttrademark           = {100,100},
5965 ⟨bch⟩      \texttrademark        = {150,150},
5966 ⟨blg | cmr | ppl⟩      \texttrademark             = {200,200},
5967 ⟨pmn⟩      \texttrademark        = {  50,  50},
5968 ⟨ugm⟩      \texttrademark        = {100,150},
5969 ⟨bch | ugm⟩      \textcent               = {  50,    },
5970 ⟨ptm⟩      \textcent             = {100,100},
5971 ⟨bch⟩      \textsterling         = {  50,    },
5972 ⟨ugm⟩      \textsterling         = {    ,  50},
5973 ⟨bch⟩      \textbrokenbar        = {200,200},
5974 ⟨blg⟩      \textbrokenbar        = {250,250},
5975 ⟨ugm⟩      \textbrokenbar        = {200,300},
5976 ⟨pmn⟩      \textasciidieresis    = {300,400},
5977 ⟨m − t | bch | cmr | pad | ptm | ugm⟩      \textcopyright             = {100,100},
5978 ⟨pmn⟩      \textcopyright        = {100,150},
5979 ⟨ppl⟩      \textcopyright        = {200,200},
5980 ⟨bch | cmr | ugm⟩      \textordfeminine         = {100,200},
5981 ⟨pad | pmn⟩      \textordfeminine        = {200,200},
5982 ⟨bch | cmr | pad | pmn | ugm⟩      \textlnot                = {200,    },
5983 ⟨blg⟩      \textlnot             = {200,100},
5984 ⟨m − t | bch | cmr | pad | ptm | ugm⟩      \textregistered          = {100,100},
5985 ⟨pmn⟩      \textregistered       = {  50,150},
5986 ⟨ppl⟩      \textregistered       = {200,200},
5987 ⟨pmn⟩      \textasciimacron      = {150,200},
5988 ⟨m − t | ppl | ptm⟩      \textdegree              = {300,300},
5989 ⟨bch⟩      \textdegree           = {150,200},
5990 ⟨blg | ugm⟩      \textdegree             = {200,200},
5991 ⟨cmr | pad⟩      \textdegree             = {400,400},
5992 ⟨pmn⟩      \textdegree           = {150,400},
5993 ⟨bch | cmr | pad | pmn | ugm⟩      \textpm                = {150,200},
5994 ⟨blg⟩      \textpm               = {100,100},
5995 ⟨ptm⟩      \textpm               = {  50,  80},
5996 ⟨bch | blg | ugm⟩      \texttwosuperior         = {100,200},
5997 ⟨cmr⟩      \texttwosuperior      = {  50,100},
5998 ⟨pad | pmn⟩      \texttwosuperior        = {200,200},
```

```
5999 ⟨ptm⟩      \texttwosuperior           = { 50, 50},
6000 ⟨bch | blg | ugm⟩      \textthreesuperior          = {100,200},
6001 ⟨cmr⟩      \textthreesuperior         = { 50,100},
6002 ⟨pad | pmn⟩      \textthreesuperior          = {200,200},
6003 ⟨ptm⟩      \textthreesuperior         = { 50, 50},
6004 ⟨pmn⟩      \textasciiacute            = {300,400},
6005 ⟨bch | ugm⟩      \textmu               = {   ,100},
6006 ⟨bch | pad | pmn⟩      \textparagraph           = {   ,100},
6007 ⟨bch | cmr | pad | pmn⟩      \textperiodcentered      = {300,400},
6008 ⟨blg⟩      \textperiodcentered        = {400,500},
6009 ⟨ptm⟩      \textperiodcentered        = {300,300},
6010 ⟨ugm⟩      \textperiodcentered        = {200,500},
6011 ⟨bch | blg | ugm⟩      \textonesuperior          = {200,300},
6012 ⟨cmr | pad | pmn⟩      \textonesuperior           = {200,200},
6013 ⟨ptm⟩      \textonesuperior           = {100,100},
6014 ⟨bch | pad | pmn | ugm⟩      \textordmasculine          = {200,200},
6015 ⟨blg | cmr⟩      \textordmasculine         = {100,200},
6016 ⟨bch | cmr | pmn⟩      \texteuro                = {100,    },
6017 ⟨pad⟩      \texteuro                  = { 50,100},
6018 ⟨bch⟩      \texttimes                 = {200,200},
6019 ⟨blg | ptm⟩      \texttimes              = {100,100},
6020 ⟨cmr⟩      \texttimes                 = {150,250},
6021 ⟨pad⟩      \texttimes                 = {100,150},
6022 ⟨pmn⟩      \texttimes                 = { 70,100},
6023 ⟨ugm⟩      \texttimes                 = {200,300},
6024 ⟨bch | pad | pmn⟩      \textdiv              = {150,200}
6025 ⟨blg⟩      \textdiv                   = {100,100}
6026 ⟨cmr⟩      \textdiv                   = {150,250}
6027 ⟨ptm⟩      \textdiv                   = { 50,100},
6028 ⟨ugm⟩      \textdiv                   = {200,300},
6029 ⟨ptm⟩      \textperthousand           = {   ,50}
6030 ⟨ugm⟩      \textsection               = {   ,100},
6031 ⟨ugm⟩      \textonehalf               = { 50,100},
6032 ⟨ugm⟩      \textonequarter            = { 50,100},
6033 ⟨ugm⟩      \textthreequarters         = { 50,100},
6034 ⟨ugm⟩      \textsurd                  = {   ,100}
```

Remaining slots in the source file.

```
6035      }
6036
6037 ⟨∗cmr | pad | pmn | ugm⟩
6038 \SetProtrusion
6039 ⟨cmr⟩      [ name      = cmr-textcomp-it ]
6040 ⟨pad⟩      [ name      = pad-textcomp-it ]
6041 ⟨pmn⟩      [ name      = pmn-textcomp-it ]
6042 ⟨ugm⟩      [ name      = ugm-textcomp-it ]
6043      { encoding = TS1,
6044 ⟨cmr⟩      family   = cmr,
6045 ⟨pad⟩      family   = {pad,padx,padj},
6046 ⟨pmn⟩      family   = {pmnx,pmnj},
6047 ⟨ugm⟩      family   = ugm,
6048 ⟨!ugm⟩      shape    = {it,sl} }
6049 ⟨ugm⟩      shape    = it }
6050      {
6051 ⟨cmr⟩      \textquotestraightbase     = {300,600},
6052 ⟨pad | pmn⟩      \textquotestraightbase     = {400,400},
6053 ⟨cmr⟩      \textquotestraightdblbase = {300,600},
6054 ⟨pad⟩      \textquotestraightdblbase = {300,400},
6055 ⟨pmn⟩      \textquotestraightdblbase = {300,300},
6056      \texttwelveudash           = {200,200},
```

```
6057 ⟨cmr | pad | pmn⟩      \textthreequartersemdash  = {150,150},
6058 ⟨ugm⟩      \textthreequartersemdash  = {200,200},
6059 ⟨cmr⟩      \textquotesingle          = {600,300},
6060 ⟨pad⟩      \textquotesingle          = {800,100},
6061 ⟨pmn⟩      \textquotesingle          = {300,200},
6062 ⟨ugm⟩      \textquotesingle          = {500,500},
6063 ⟨cmr⟩      \textasteriskcentered     = {300,200},
6064 ⟨pad⟩      \textasteriskcentered     = {500,100},
6065 ⟨pmn⟩      \textasteriskcentered     = {200,300},
6066 ⟨ugm⟩      \textasteriskcentered     = {300,150},
6067 ⟨pmn⟩      \textfractionsolidus      = {-200,-200},
6068 ⟨cmr⟩      \textoneoldstyle          = {100, 50},
6069 ⟨pad⟩      \textoneoldstyle          = {100,    },
6070 ⟨pmn⟩      \textoneoldstyle          = { 50,    },
6071 ⟨pad⟩      \texttwooldstyle          = { 50,    },
6072 ⟨pmn⟩      \texttwooldstyle          = {-50,    },
6073 ⟨cmr⟩      \textthreeoldstyle        = {100, 50},
6074 ⟨pmn⟩      \textthreeoldstyle        = {-100,   },
6075 ⟨cmr⟩      \textfouroldstyle         = { 50, 50},
6076 ⟨pad⟩      \textfouroldstyle         = { 50,100},
6077 ⟨cmr⟩      \textsevenoldstyle        = { 50, 80},
6078 ⟨pad⟩      \textsevenoldstyle        = { 50,    },
6079 ⟨pmn⟩      \textsevenoldstyle        = { 20,    },
6080 ⟨cmr⟩      \textlangle               = {400,    },
6081 ⟨cmr⟩      \textrangle               = {    ,400},
6082 ⟨cmr | pad⟩      \textminus               = {300,300},
6083 ⟨pmn⟩      \textminus                = {200,200},
6084 ⟨ugm⟩      \textminus                = {250,300},
6085 ⟨pad | pmn⟩      \textlbrackdbl           = {100,    },
6086 ⟨pad | pmn⟩      \textrbrackdbl           = {    ,100},
6087 ⟨pmn⟩      \textasciigrave           = {300,300},
6088 ⟨cmr | pad | pmn⟩      \texttildelow             = {200,250},
6089 ⟨pmn⟩      \textasciibreve           = {300,300},
6090 ⟨pmn⟩      \textasciicaron           = {300,300},
6091 ⟨pmn⟩      \textacutedbl             = {200,300},
6092 ⟨pmn⟩      \textgravedbl             = {150,300},
6093 ⟨cmr⟩      \textdagger               = {100,100},
6094 ⟨pad⟩      \textdagger               = {200,100},
6095 ⟨pmn⟩      \textdagger               = { 80, 50},
6096 ⟨ugm⟩      \textdagger               = { 80, 80},
6097 ⟨cmr | pad⟩      \textdaggerdbl           = { 80, 80},
6098 ⟨pmn⟩      \textdaggerdbl            = { 80, 50},
6099 ⟨ugm⟩      \textbardbl               = {150,150},
6100 ⟨cmr⟩      \textbullet               = {200,100},
6101 ⟨pad⟩      \textbullet               = {300,    },
6102 ⟨pmn⟩      \textbullet               = { 30, 70},
6103 ⟨ugm⟩      \textbullet               = { 50,100},
6104 ⟨cmr⟩      \textcelsius              = {100,    },
6105 ⟨pad⟩      \textcelsius              = {200,    },
6106 ⟨pmn⟩      \textcelsius              = { 50,-50},
6107 ⟨pad⟩      \textflorin               = {100,    },
6108 ⟨pmn⟩      \textflorin               = { 50,100},
6109 ⟨ugm⟩      \textflorin               = {    ,100},
6110 ⟨cmr⟩      \textcolonmonetary        = {150,    },
6111 ⟨pad⟩      \textcolonmonetary        = {100,    },
6112 ⟨pmn⟩      \textcolonmonetary        = { 50,-50},
6113 ⟨cmr | pad⟩      \texttrademark           = {200,    },
6114 ⟨pmn⟩      \texttrademark            = { 50,100},
6115 ⟨ugm⟩      \texttrademark            = {150, 50},
6116 ⟨ugm⟩      \textcent                 = { 50,    },
```

```
6117 ⟨ugm⟩      \textsterling           = {    , 50},
6118 ⟨ugm⟩      \textbrokenbar          = {200,300},
6119 ⟨pmn⟩      \textasciidieresis      = {300,200},
6120 ⟨cmr⟩      \textcopyright          = {100,    },
6121 ⟨pad⟩      \textcopyright          = {200,100},
6122 ⟨pmn⟩      \textcopyright          = {100,150},
6123 ⟨ugm⟩      \textcopyright          = {300,    },
6124 ⟨cmr⟩      \textordfeminine        = {100,100},
6125 ⟨pmn⟩      \textordfeminine        = {200,200},
6126 ⟨ugm⟩      \textordfeminine        = {100,200},
6127 ⟨cmr | pad⟩     \textlnot               = {300,    },
6128 ⟨pmn | ugm⟩     \textlnot               = {200,    },
6129 ⟨cmr⟩      \textregistered         = {100,    },
6130 ⟨pad⟩      \textregistered         = {200,100},
6131 ⟨pmn⟩      \textregistered         = { 50,150},
6132 ⟨ugm⟩      \textregistered         = {300,    },
6133 ⟨pmn⟩      \textasciimacron        = {150,200},
6134 ⟨cmr | pad⟩     \textdegree             = {500,100},
6135 ⟨pmn⟩      \textdegree             = {150,150},
6136 ⟨ugm⟩      \textdegree             = {300,200},
6137 ⟨cmr⟩      \textpm                 = {150,100},
6138 ⟨pad⟩      \textpm                 = {200,150},
6139 ⟨pmn | ugm⟩     \textpm                 = {150,200},
6140 ⟨cmr⟩      \textonesuperior        = {400,    },
6141 ⟨pad⟩      \textonesuperior        = {300,100},
6142 ⟨pmn⟩      \textonesuperior        = {200,100},
6143 ⟨ugm⟩      \textonesuperior        = {300,300},
6144 ⟨cmr⟩      \texttwosuperior        = {400,    },
6145 ⟨pad⟩      \texttwosuperior        = {300,    },
6146 ⟨pmn⟩      \texttwosuperior        = {200,100},
6147 ⟨ugm⟩      \texttwosuperior        = {300,200},
6148 ⟨cmr⟩      \textthreesuperior      = {400,    },
6149 ⟨pad⟩      \textthreesuperior      = {300,    },
6150 ⟨pmn⟩      \textthreesuperior      = {200,100},
6151 ⟨ugm⟩      \textthreesuperior      = {300,200},
6152 ⟨ugm⟩      \textmu                 = {    ,100},
6153 ⟨pmn⟩      \textasciiacute         = {300,200},
6154 ⟨cmr⟩      \textparagraph          = {200,    },
6155 ⟨pmn⟩      \textparagraph          = {    ,100},
6156 ⟨cmr⟩      \textperiodcentered     = {500,500},
6157 ⟨pad | pmn | ugm⟩     \textperiodcentered     = {300,400},
6158 ⟨cmr⟩      \textordmasculine       = {100,100},
6159 ⟨pmn⟩      \textordmasculine       = {200,200},
6160 ⟨ugm⟩      \textordmasculine       = {300,200},
6161 ⟨cmr⟩      \texteuro               = {200,    },
6162 ⟨pad⟩      \texteuro               = {100,    },
6163 ⟨pmn⟩      \texteuro               = {100,-50},
6164 ⟨cmr⟩      \texttimes              = {200,200},
6165 ⟨pad⟩      \texttimes              = {200,100},
6166 ⟨pmn⟩      \texttimes              = { 70,100},
6167 ⟨ugm⟩      \texttimes              = {200,300},
6168 ⟨cmr | pad⟩     \textdiv                = {200,200}
6169 ⟨pmn⟩      \textdiv                = {150,200}
6170 ⟨ugm⟩      \textdiv                = {200,300},
6171 ⟨ugm⟩      \textsection            = {    ,200},
6172 ⟨ugm⟩      \textonehalf            = { 50,100},
6173 ⟨ugm⟩      \textonequarter         = { 50,100},
6174 ⟨ugm⟩      \textthreequarters      = { 50,100},
6175 ⟨ugm⟩      \textsurd               = {    ,100}
6176    }
```

6177
6178 ⟨/cmr | pad | pmn | ugm⟩

15.8.6 Computer Modern math

Now to the math symbols for Computer Modern Roman. Definitions have been extracted from `fontmath.ltx`. I did not spend too much time fiddling with these settings, so they can surely be improved.

The math font 'operators' (also used for the \mathrm and \mathbf alphabets) is OT1/cmr, which we've already set up above. It's declared as:

```
\DeclareSymbolFont{operators}  {OT1}{cmr}{m} {n}
\SetSymbolFont{operators}{bold}{OT1}{cmr}{bx}{n}
```

\mathit (OT1/cmr/m/it) is also already set up.

There are (for the moment) no settings for \mathsf and \mathtt.

Math font 'letters' (also used as \mathnormal) is declared as:

```
\DeclareSymbolFont{letters}    {OML}{cmm}{m}{it}
\SetSymbolFont{letters}  {bold}{OML}{cmm}{b}{it}
```

6179 ⟨∗cmr⟩
6180 \SetProtrusion
6181   [name   = cmr-math-letters]
6182   { encoding = OML,
6183   family   = cmm,
6184   series   = {m,b},
6185   shape   = it   }
6186   {
6187   A = {100, 50}, % \mathnormal
6188   B = { 50,   },
6189   C = { 50,   },
6190   D = { 50, 50},
6191   E = { 50,   },
6192   F = {100, 50},
6193   G = { 50, 50},
6194   H = { 50, 50},
6195   I = { 50, 50},
6196   J = {150, 50},
6197   K = { 50,100},
6198   L = { 50, 50},
6199   M = { 50,   },
6200   N = { 50,   },
6201   O = { 50,   },
6202   P = { 50,   },
6203   Q = { 50, 50},
6204   R = { 50,   },
6205   S = { 50,   },
6206   T = { 50,100},
6207   U = { 50, 50},
6208   V = {100,100},
6209   W = { 50,100},
6210   X = { 50,100},
6211   Y = {100,100},
6212   f = {100,100},
6213   h = {   ,100},
6214   i = {   , 50},
6215   j = {   , 50},

```
6216          k = {    , 50},
6217          r = {    , 50},
6218          v = {    , 50},
6219          w = {    , 50},
6220          x = {    , 50},
6221        "0B = { 50,100}, % \alpha
6222        "0C = { 50, 50}, % \beta
6223        "0D = {200,150}, % \gamma
6224        "0E = { 50, 50}, % \delta
6225        "0F = { 50, 50}, % \epsilon
6226        "10 = { 50,150}, % \zeta
6227        "12 = { 50,    }, % \theta
6228        "13 = {    ,100}, % \iota
6229        "14 = {    ,100}, % \kappa
6230        "15 = {100, 50}, % \lambda
6231        "16 = {    , 50}, % \mu
6232        "17 = {    , 50}, % \nu
6233        "18 = {    , 50}, % \xi
6234        "19 = { 50,100}, % \pi
6235        "1A = { 50, 50}, % \rho
6236        "1B = {    ,150}, % \sigma
6237        "1C = { 50,150}, % \tau
6238        "1D = { 50, 50}, % \upsilon
6239        "1F = { 50,100}, % \chi
6240        "20 = { 50, 50}, % \psi
6241        "21 = {    , 50}, % \omega
6242        "22 = {    , 50}, % \varepsilon
6243        "23 = {    , 50}, % \vartheta
6244        "24 = {    , 50}, % \varpi
6245        "25 = {100,    }, % \varrho
6246        "26 = {100,100}, % \varsigma
6247        "27 = { 50, 50}, % \varphi
6248        "28 = {100,100}, % \leftharpoonup
6249        "29 = {100,100}, % \leftharpoondown
6250        "2A = {100,100}, % \rightharpoonup
6251        "2B = {100,100}, % \rightharpoondown
6252        "2C = {300,200}, % \lhook
6253        "2D = {200,300}, % \rhook
6254        "2E = {    ,100}, % \triangleright
6255        "2F = {100,    }, % \triangleleft
6256        "3A = {    ,500}, % ., \ldotp
6257        "3B = {    ,500}, % ,
6258        "3C = {200,100}, % <
6259        "3D = {300,400}, % /
6260        "3E = {100,200}, % >
6261        "3F = {200,200}, % \star
6262        "5B = {    ,100}, % \flat
6263        "5E = {200,200}, % \smile
6264        "5F = {200,200}, % \frown
6265        "7C = {100,    }, % \jmath
6266        "7D = {    ,100}  % \wp
```

Remaining slots in the source file.

```
6267    }
6268
```

Math font 'symbols' (also used for the \mathcal alphabet) is declared as:

```
\DeclareSymbolFont{symbols}    {OMS}{cmsy}{m}{n}
\SetSymbolFont{symbols}  {bold}{OMS}{cmsy}{b}{n}
```

```
6269  \SetProtrusion
6270    [ name      = cmr-math-symbols ]
6271    { encoding = OMS,
6272      family   = cmsy,
6273      series   = {m,b},
6274      shape    = n  }
6275    {
6276       A = {150, 50}, % \mathcal
6277       C = {   ,100},
6278       D = {   , 50},
6279       F = { 50,150},
6280       I = {   ,100},
6281       J = {100,150},
6282       K = {   ,100},
6283       L = {100,   },
6284       M = { 50, 50},
6285       N = { 50,100},
6286       P = {   , 50},
6287       Q = { 50,   },
6288       R = {   , 50},
6289       T = { 50,150},
6290       V = { 50, 50},
6291       W = {   , 50},
6292       X = {100,100},
6293       Y = {100,   },
6294       Z = {100,150},
6295      "00 = {300,300}, % -
6296      "01 = {   ,700}, % \cdot, \cdotp
6297      "02 = {150,250}, % \times
6298      "03 = {150,250}, % *, \ast
6299      "04 = {200,300}, % \div
6300      "05 = {150,250}, % \diamond
6301      "06 = {200,200}, % \pm
6302      "07 = {200,200}, % \mp
6303      "08 = {100,100}, % \oplus
6304      "09 = {100,100}, % \ominus
6305      "0A = {100,100}, % \otimes
6306      "0B = {100,100}, % \oslash
6307      "0C = {100,100}, % \odot
6308      "0D = {100,100}, % \bigcirc
6309      "0E = {100,100}, % \circ
6310      "0F = {100,100}, % \bullet
6311      "10 = {100,100}, % \asymp
6312      "11 = {100,100}, % \equiv
6313      "12 = {200,100}, % \subseteq
6314      "13 = {100,200}, % \supseteq
6315      "14 = {200,100}, % \leq
6316      "15 = {100,200}, % \geq
6317      "16 = {200,100}, % \preceq
6318      "17 = {100,200}, % \succeq
6319      "18 = {200,200}, % \sim
6320      "19 = {150,150}, % \approx
6321      "1A = {200,100}, % \subset
6322      "1B = {100,200}, % \supset
6323      "1C = {200,100}, % \ll
6324      "1D = {100,200}, % \gg
6325      "1E = {300,100}, % \prec
6326      "1F = {100,300}, % \succ
6327      "20 = {100,200}, % \leftarrow
6328      "21 = {200,100}, % \rightarrow
```

```
6329        "22 = {100,100}, % \uparrow
6330        "23 = {100,100}, % \downarrow
6331        "24 = {100,100}, % \leftrightarrow
6332        "25 = {100,100}, % \nearrow
6333        "26 = {100,100}, % \searrow
6334        "27 = {100,100}, % \simeq
6335        "28 = {100,100}, % \Leftarrow
6336        "29 = {100,100}, % \Rightarrow
6337        "2A = {100,100}, % \Uparrow
6338        "2B = {100,100}, % \Downarrow
6339        "2C = {100,100}, % \Leftrightarrow
6340        "2D = {100,100}, % \nwarrow
6341        "2E = {100,100}, % \swarrow
6342        "2F = {    ,100}, % \propto
6343        "30 = {    ,400}, % \prime
6344        "31 = {100,100}, % \infty
6345        "32 = {150,100}, % \in
6346        "33 = {100,150}, % \ni
6347        "34 = {100,100}, % \triangle, \bigtriangleup
6348        "35 = {100,100}, % \bigtriangledown
6349        "38 = {    ,100}, % \forall
6350        "39 = {100,    }, % \exists
6351        "3A = {200,    }, % \neg
6352        "3E = {200,200}, % \top
6353        "3F = {200,200}, % \bot, \perp
6354        "5E = {100,200}, % \wedge
6355        "5F = {100,200}, % \vee
6356        "60 = {    ,300}, % \vdash
6357        "61 = {300,    }, % \dashv
6358        "62 = {100,100}, % \lfloor
6359        "63 = {100,100}, % \rfloor
6360        "64 = {100,100}, % \lceil
6361        "65 = {100,100}, % \rceil
6362        "66 = {150,    }, % \lbrace
6363        "67 = {    ,150}, % \rbrace
6364        "68 = {400,    }, % \langle
6365        "69 = {    ,400}, % \rangle
6366        "6C = {100,100}, % \updownarrow
6367        "6D = {100,100}, % \Updownarrow
6368        "6E = {100,300}, % \, \backslash, \setminus
6369        "72 = {100,100}, % \nabla
6370        "79 = {200,200}, % \dagger
6371        "7A = {100,100}, % \ddagger
6372        "7B = {100,    }, % \mathparagraph
6373        "7C = {100,100}, % \clubsuit
6374        "7D = {100,100}, % \diamondsuit
6375        "7E = {100,100}, % \heartsuit
6376        "7F = {100,100}  % \spadesuit
```

Remaining slots in the source file.

```
6377    }
6378
```

We don't bother about 'largesymbols', since it will only be used in display math, where protrusion doesn't work anyway. It's declared as:

```
\DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}
```

```
6379 ⟨/cmr⟩
6380 ⟨/cfg − t⟩
```

15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
6381  ⟨∗cfg − u⟩
```

Symbol font 'a'.

```
6382  ⟨∗msa⟩
6383  \SetProtrusion
6384    [ name     = AMS-a ]
6385    { encoding = U,
6386      family   = msa  }
6387    {
6388      "05  = {150,250},  % \centerdot
6389      "06  = {100,100},  % \lozenge
6390      "07  = { 50, 50},  % \blacklozenge
6391      "08  = { 50, 50},  % \circlearrowright
6392      "09  = { 50, 50},  % \circlearrowleft
6393      "0A  = {100,100},  % \rightleftharpoons
6394      "0B  = {100,100},  % \leftrightharpoons
6395      "0D  = {-50,200},  % \Vdash
6396      "0E  = {-50,200},  % \Vvdash
6397      "0F  = {-70,150},  % \vDash
6398      "10  = {100,150},  % \twoheadrightarrow
6399      "11  = {100,150},  % \twoheadleftarrow
6400      "12  = { 50,100},  % \leftleftarrows
6401      "13  = { 50, 80},  % \rightrightarrows
6402      "14  = {120,120},  % \upuparrows
6403      "15  = {120,120},  % \downdownarrows
6404      "16  = {200,200},  % \upharpoonright
6405      "17  = {200,200},  % \downharpoonright
6406      "18  = {200,200},  % \upharpoonleft
6407      "19  = {200,200},  % \downharpoonleft
6408      "1A  = { 80,100},  % \rightarrowtail
6409      "1B  = { 80,100},  % \leftarrowtail
6410      "1C  = { 50, 50},  % \leftrightarrows
6411      "1D  = { 50, 50},  % \rightleftarrows
6412      "1E  = {250,   },  % \Lsh
6413      "1F  = {   ,250},  % \Rsh
6414      "20  = {100,100},  % \rightsquigarrow
6415      "21  = {100,100},  % \leftrightsquigarrow
6416      "22  = {100, 50},  % \looparrowleft
6417      "23  = { 50,100},  % \looparrowright
6418      "24  = { 50, 80},  % \circeq
6419      "25  = {   ,100},  % \succsim
6420      "26  = {   ,100},  % \gtrsim
6421      "27  = {   ,100},  % \gtrapprox
6422      "28  = {150, 50},  % \multimap
6423      "2B  = {100,150},  % \doteqdot
6424      "2C  = {100,150},  % \triangleq
6425      "2D  = {100, 50},  % \precsim
6426      "2E  = {100, 50},  % \lesssim
6427      "2F  = { 50, 50},  % \lessapprox
6428      "30  = {100, 50},  % \eqslantless
6429      "31  = { 50, 50},  % \eqslantgtr
6430      "32  = {100, 50},  % \curlyeqprec
6431      "33  = { 50,100},  % \curlyeqsucc
6432      "34  = {100, 50},  % \preccurlyeq
6433      "36  = { 50,   },  % \leqslant
6434      "38  = {   , 50},  % \backprime
6435      "39  = {250,250},  % \dabar@ : the dash bar in \dash(left,right)arrow
```

```
6436        "3C  =  {  50,100},   %  \succcurlyeq
6437        "3E  =  {    ,  50},   %  \geqslant
6438        "40  =  {    ,  50},   %  \sqsubset
6439        "41  =  {  50,    },   %  \sqsupset
6440        "42  =  {    ,150},    %  \vartriangleright, \rhd
6441        "43  =  {150,    },    %  \vartriangleleft, \lhd
6442        "44  =  {    ,100},    %  \trianglerighteq, \unrhd
6443        "45  =  {100,    },    %  \trianglelefteq, \unlhd
6444        "46  =  {100,100},     %  \bigstar
6445        "48  =  {  50,  50},   %  \blacktriangledown
6446        "49  =  {    ,100},    %  \blacktriangleright
6447        "4A  =  {100,    },    %  \blacktriangleleft
6448        "4B  =  {    ,150},    %  \dashrightarrow (the arrow)
6449        "4C  =  {150,    },    %  \dashleftarrow
6450        "4D  =  {  50,  50},   %  \vartriangle
6451        "4E  =  {  50,  50},   %  \blacktriangle
6452        "4F  =  {  50,  50},   %  \triangledown
6453        "50  =  {  50,  50},   %  \eqcirc
6454        "56  =  {    ,150},    %  \Rrightarrow
6455        "57  =  {150,    },    %  \Lleftarrow
6456        "58  =  {100,300},     %  \checkmark
6457        "5C  =  {  50,  50},   %  \angle
6458        "5D  =  {  50,  50},   %  \measuredangle
6459        "5E  =  {  50,  50},   %  \sphericalangle
6460        "5F  =  {    ,  50},   %  \varpropto
6461        "60  =  {100,100},     %  \smallsmile
6462        "61  =  {100,100},     %  \smallfrown
6463        "62  =  {  50,    },   %  \Subset
6464        "63  =  {    ,  50},   %  \Supset
6465        "66  =  {150,150},     %  \curlywedge
6466        "67  =  {150,150},     %  \curlyvee
6467        "68  =  {  50,150},    %  \leftthreetimes
6468        "69  =  {100,  50},    %  \rightthreetimes
6469        "6C  =  {  50,  50},   %  \bumpeq
6470        "6D  =  {  50,  50},   %  \Bumpeq
6471        "6E  =  {100,    },    %  \lll
6472        "6F  =  {    ,100},    %  \ggg
6473        "70  =  {  50,100},    %  \ulcorner
6474        "71  =  {100,  50},    %  \urcorner
6475        "75  =  {150,200},     %  \dotplus
6476        "76  =  {  50,100},    %  \backsim
6477        "78  =  {  50,100},    %  \llcorner
6478        "79  =  {100,  50},    %  \lrcorner
6479        "7C  =  {100,100},     %  \intercal
6480        "7D  =  {  50,  50},   %  \circledcirc
6481        "7E  =  {  50,  50},   %  \circledast
6482        "7F  =  {  50,  50}    %  \circleddash
```

Remaining slots in the source file.

```
6483      }
6484
6485  ⟨/msa⟩
```

Symbol font 'b'.

```
6486  ⟨*msb⟩
6487  \SetProtrusion
6488    [ name      = AMS-b ]
6489    { encoding = U,
6490      family   = msb  }
6491    {
```

```
6492        A   =  {  50,  50},  %  \mathbb
6493        C   =  {  50,  50},
6494        G   =  {    ,  50},
6495        L   =  {    ,  50},
6496        P   =  {    ,  50},
6497        R   =  {    ,  50},
6498        T   =  {    ,  50},
6499        V   =  {  50,  50},
6500        X   =  {  50,  50},
6501        Y   =  {  50,  50},
6502       "00  =  {  50,  50},  %  \lvertneqq
6503       "01  =  {  50,  50},  %  \gvertneqq
6504       "02  =  {  50,  50},  %  \nleq
6505       "03  =  {  50,  50},  %  \ngeq
6506       "04  =  {100,  50},  %  \nless
6507       "05  =  {  50,150},  %  \ngtr
6508       "06  =  {100,  50},  %  \nprec
6509       "07  =  {  50,150},  %  \nsucc
6510       "08  =  {  50,  50},  %  \lneqq
6511       "09  =  {  50,  50},  %  \gneqq
6512       "0A  =  {100,100},  %  \nleqslant
6513       "0B  =  {100,100},  %  \ngeqslant
6514       "0C  =  {100,  50},  %  \lneq
6515       "0D  =  {  50,100},  %  \gneq
6516       "0E  =  {100,  50},  %  \npreceq
6517       "0F  =  {  50,100},  %  \nsucceq
6518       "10  =  {  50,    },  %  \precnsim
6519       "11  =  {  50,  50},  %  \succnsim
6520       "12  =  {  50,  50},  %  \lnsim
6521       "13  =  {  50,  50},  %  \gnsim
6522       "14  =  {  50,  50},  %  \nleqq
6523       "15  =  {  50,  50},  %  \ngeqq
6524       "16  =  {  50,  50},  %  \precneqq
6525       "17  =  {  50,  50},  %  \succneqq
6526       "18  =  {  50,  50},  %  \precnapprox
6527       "19  =  {  50,  50},  %  \succnapprox
6528       "1A  =  {  50,  50},  %  \lnapprox
6529       "1B  =  {  50,  50},  %  \gnapprox
6530       "1C  =  {150,200},  %  \nsim
6531       "1D  =  {  50,  50},  %  \ncong
6532       "1E  =  {100,150},  %  \diagup
6533       "1F  =  {100,150},  %  \diagdown
6534       "20  =  {100,  50},  %  \varsubsetneq
6535       "21  =  {  50,100},  %  \varsupsetneq
6536       "22  =  {100,  50},  %  \nsubseteqq
6537       "23  =  {  50,100},  %  \nsupseteqq
6538       "24  =  {100,  50},  %  \subsetneqq
6539       "25  =  {  50,100},  %  \supsetneqq
6540       "26  =  {100,  50},  %  \varsubsetneqq
6541       "27  =  {  50,100},  %  \varsupsetneqq
6542       "28  =  {100,  50},  %  \subsetneq
6543       "29  =  {  50,100},  %  \supsetneq
6544       "2A  =  {100,  50},  %  \nsubseteq
6545       "2B  =  {  50,100},  %  \nsupseteq
6546       "2C  =  {  50,100},  %  \nparallel
6547       "2D  =  {100,150},  %  \nmid
6548       "2E  =  {150,150},  %  \nshortmid
6549       "2F  =  {100,100},  %  \nshortparallel
6550       "30  =  {    ,150},  %  \nvdash
6551       "31  =  {    ,150},  %  \nVdash
```

```
6552      "32  =  {    ,100},   % \nvDash
6553      "33  =  {    ,100},   % \nVDash
6554      "34  =  {    ,100},   % \ntrianglerighteq
6555      "35  =  {100,    },   % \ntrianglelefteq
6556      "36  =  {100,    },   % \ntriangleleft
6557      "37  =  {    ,100},   % \ntriangleright
6558      "38  =  {100,200},   % \nleftarrow
6559      "39  =  {100,200},   % \nrightarrow
6560      "3A  =  {100,100},   % \nLeftarrow
6561      "3B  =  { 50,100},   % \nRightarrow
6562      "3C  =  {100,100},   % \nLeftrightarrow
6563      "3D  =  {100,200},   % \nleftrightarrow
6564      "3E  =  { 50, 50},   % \divideontimes
6565      "3F  =  { 50, 50},   % \varnothing
6566      "60  =  {200,    },   % \Finv
6567      "61  =  {    , 50},   % \Game
6568      "68  =  {100,100},   % \eqsim
6569      "69  =  { 50,    },   % \beth
6570      "6A  =  { 50,    },   % \gimel
6571      "6B  =  {150,    },   % \daleth
6572      "6C  =  {200,    },   % \lessdot
6573      "6D  =  {    ,200},   % \gtrdot
6574      "6E  =  {100,200},   % \ltimes
6575      "6F  =  {150,100},   % \rtimes
6576      "70  =  { 50,100},   % \shortmid
6577      "71  =  { 50, 50},   % \shortparallel
6578      "72  =  {200,300},   % \smallsetminus
6579      "73  =  {100,200},   % \thicksim
6580      "74  =  { 50,100},   % \thickapprox
6581      "75  =  { 50, 50},   % \approxeq
6582      "76  =  { 50,100},   % \succapprox
6583      "77  =  { 50, 50},   % \precapprox
6584      "78  =  {100,100},   % \curvearrowleft
6585      "79  =  { 50,150},   % \curvearrowright
6586      "7A  =  { 50,200},   % \digamma
6587      "7B  =  {100, 50},   % \varkappa
6588      "7F  =  {200,    }    % \backepsilon
```

Remaining slots in the source file.

```
6589      }
6590
6591  ⟨/msb⟩
```

15.8.8   Euler

Euler Roman font (package euler).

```
6592  ⟨*eur⟩
6593  \SetProtrusion
6594     [ name     = euler ]
6595     { encoding = U,
6596       family   = eur  }
6597     {
6598       "01  =  {100,100},
6599       "03  =  {100,150},
6600       "06  =  {    ,100},
6601       "07  =  {100,150},
6602       "08  =  {100,100},
6603       "0A  =  {100,100},
6604       "0B  =  {    , 50},
```

```
6605      "0C  =  {    ,100},
6606      "0D  =  {100,100},
6607      "0E  =  {    ,100},
6608      "0F  =  {100,100},
6609      "10  =  {100,100},
6610      "13  =  {    ,100},
6611      "14  =  {    ,100},
6612      "15  =  {    , 50},
6613      "16  =  {    , 50},
6614      "17  =  { 50,100},
6615      "18  =  { 50,100},
6616      "1A  =  {    , 50},
6617      "1B  =  {    , 50},
6618      "1C  =  { 50,100},
6619      "1D  =  { 50,100},
6620      "1E  =  { 50,100},
6621      "1F  =  { 50,100},
6622      "20  =  {    , 50},
6623      "21  =  {    , 50},
6624      "22  =  { 50,100},
6625      "24  =  {    , 50},
6626      "27  =  { 50,100},
6627       1   =  {100,100},
6628       7   =  { 50,100},
6629      "3A  =  {300,500},
6630      "3B  =  {200,400},
6631      "3C  =  {200,100},
6632      "3D  =  {200,200},
6633      "3E  =  {100,200},
6634       A   =  {    ,100},
6635       D   =  {    , 50},
6636       J   =  { 50,   },
6637       K   =  {    , 50},
6638       L   =  {    , 50},
6639       Q   =  {    , 50},
6640       T   =  { 50,   },
6641       X   =  { 50, 50},
6642       Y   =  { 50,   },
6643       h   =  {    , 50},
6644       k   =  {    , 50}
6645    }
6646
```

Extended by the eulervm package.

```
6647 \SetProtrusion
6648    [ name     = euler-vm,
6649      load     = euler ]
6650    { encoding = U,
6651      family   = zeur  }
6652    {
6653      "28  =  {100,200},
6654      "29  =  {100,200},
6655      "2A  =  {100,150},
6656      "2B  =  {100,150},
6657      "2C  =  {200,300},
6658      "2D  =  {200,300},
6659      "2E  =  {    ,100},
6660      "2F  =  {100,   },
6661      "3F  =  {150,150},
6662      "5B  =  {    ,100},
```

```
6663      "5E  =  {100,100},
6664      "5F  =  {100,100},
6665      "80  =  {    , 50},
6666      "81  =  {200,250},
6667      "82  =  {100,200}
6668    }
6669
6670  ⟨/eur⟩
```

Euler Script font (eucal).

```
6671  ⟨*eus⟩
6672  \SetProtrusion
6673    [ name      = euscript ]
6674    { encoding = U,
6675      family   = eus  }
6676    {
6677       A  =  {100,100},
6678       B  =  { 50,100},
6679       C  =  { 50, 50},
6680       D  =  { 50,100},
6681       E  =  { 50,100},
6682       F  =  { 50,    },
6683       G  =  { 50,    },
6684       H  =  {    ,100},
6685       K  =  {    , 50},
6686       L  =  {    ,150},
6687       M  =  {    , 50},
6688       N  =  {    , 50},
6689       O  =  { 50, 50},
6690       P  =  { 50, 50},
6691       T  =  {    ,100},
6692       U  =  {    , 50},
6693       V  =  { 50, 50},
6694       W  =  { 50, 50},
6695       X  =  { 50, 50},
6696       Y  =  { 50,    },
6697       Z  =  { 50,100},
6698     "00  =  {250,250},
6699     "18  =  {200,200},
6700     "3A  =  {200,150},
6701     "40  =  {    ,100},
6702     "5E  =  {100,100},
6703     "5F  =  {100,100},
6704     "66  =  { 50,    },
6705     "67  =  {    , 50},
6706     "6E  =  {200,200}
6707    }
6708
6709  \SetProtrusion
6710    [ name      = euscript-vm,
6711      load      = euscript ]
6712    { encoding = U,
6713      family   = zeus  }
6714    {
6715     "01  =  {600,600},
6716     "02  =  {200,200},
6717     "03  =  {200,200},
6718     "04  =  {200,200},
6719     "05  =  {150,150},
6720     "06  =  {200,200},
```

```
6721        "07  =  {200,200},
6722        "08  =  {100,100},
6723        "09  =  {100,100},
6724        "0A  =  {100,100},
6725        "0B  =  {100,100},
6726        "0C  =  {100,100},
6727        "0D  =  {100,100},
6728        "0E  =  {150,150},
6729        "0F  =  {100,100},
6730        "10  =  {150,150},
6731        "11  =  {100,100},
6732        "12  =  {150,100},
6733        "13  =  {100,150},
6734        "14  =  {150,100},
6735        "15  =  {100,150},
6736        "16  =  {200,100},
6737        "17  =  {100,200},
6738        "19  =  {150,150},
6739        "1A  =  {150,100},
6740        "1B  =  {100,150},
6741        "1C  =  {100,100},
6742        "1D  =  {100,100},
6743        "1E  =  {250,100},
6744        "1F  =  {100,250},
6745        "20  =  {150,200},
6746        "21  =  {150,200},
6747        "22  =  {150,150},
6748        "23  =  {150,150},
6749        "24  =  {100,200},
6750        "25  =  {150,150},
6751        "26  =  {150,150},
6752        "27  =  {100,100},
6753        "28  =  {100,100},
6754        "29  =  {100,150},
6755        "2A  =  {100,100},
6756        "2B  =  {100,100},
6757        "2C  =  {100,100},
6758        "2D  =  {150,150},
6759        "2E  =  {150,150},
6760        "2F  =  {100,100},
6761        "30  =  {100,100},
6762        "31  =  {100,100},
6763        "32  =  {100,100},
6764        "33  =  {100,100},
6765        "34  =  {100,100},
6766        "35  =  {100,100},
6767        "3E  =  {150,150},
6768        "3F  =  {150,150},
6769        "60  =  {    ,200},
6770        "61  =  {200,    },
6771        "62  =  {100,100},
6772        "63  =  {100,100},
6773        "64  =  {100,100},
6774        "65  =  {100,100},
6775        "68  =  {300,    },
6776        "69  =  {    ,300},
6777        "6C  =  {100,100},
6778        "6D  =  {100,100},
6779        "6F  =  {100,100},
6780        "72  =  {100,100},
```

```
6781        "73  =  {200,100},
6782        "76  =  {   ,100},
6783        "77  =  {100,   },
6784        "78  =  { 50, 50},
6785        "79  =  {100,100},
6786        "7A  =  {100,100},
6787        "7D  =  {150,150},
6788        "7E  =  {100,100},
6789        "A8  =  {100,100},
6790        "A9  =  {100,100},
6791        "AB  =  {200,200},
6792        "BA  =  {   ,200},
6793        "BB  =  {   ,200},
6794        "BD  =  {200,200},
6795        "DE  =  {200,200}
6796      }
6797
6798  ⟨/eus⟩
```

Euler Fraktur font (eufrak).

```
6799  ⟨∗euf⟩
6800  \SetProtrusion
6801     [ name      = mathfrak ]
6802     { encoding = U,
6803       family   = euf  }
6804     {
6805        A   =  {   , 50},
6806        B   =  {   , 50},
6807        C   =  { 50, 50},
6808        D   =  {   , 80},
6809        E   =  { 50,   },
6810        G   =  {   , 50},
6811        L   =  {   , 80},
6812        O   =  {   , 50},
6813        T   =  {   , 80},
6814        X   =  { 80, 50},
6815        Z   =  { 80, 50},
6816        b   =  {   , 50},
6817        c   =  {   , 50},
6818        k   =  {   , 50},
6819        p   =  {   , 50},
6820        q   =  { 50,   },
6821        v   =  {   , 50},
6822        w   =  {   , 50},
6823        x   =  {   , 50},
6824        1   =  {100,100},
6825        2   =  { 80, 80},
6826        3   =  { 80, 50},
6827        4   =  { 80, 50},
6828        7   =  { 50, 50},
6829       "12  =  {500,500},
6830       "13  =  {500,500},
6831        !   =  {   ,200},
6832        '   =  {200,300},
6833        (   =  {200,   },
6834        )   =  {   ,200},
6835        *   =  {200,200},
6836        +   =  {200,250},
6837        -   =  {200,200},
6838       {,} =  {300,300},
```

```
6839          .  =  {400,400},
6840        {=} =  {200,200},
6841          :  =  {    ,200},
6842          ;  =  {    ,200},
6843          ]  =  {    ,200}
6844     }
6845
6846  ⟨/euf⟩
6847  ⟨/cfg − u⟩
```

15.8.9   Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym[22]).

```
6848  ⟨∗cfg − e⟩
6849  \SetProtrusion
6850  ⟨zpeu | euroitc⟩   { encoding = U,
6851  ⟨mvs⟩    { encoding = {OT1,U},
6852  ⟨zpeu⟩       family   = zpeu }
6853  ⟨euroitc⟩      family   = {euroitc,euroitcs} }
6854  ⟨mvs⟩       family   = mvs }
6855     {
6856  ⟨zpeu⟩       E = {50, }
6857  ⟨euroitc⟩      E = {100,50}
6858  ⟨mvs⟩       164 = {50,50},    % \EUR
6859  ⟨mvs⟩       068 = {50,-100}  % \EURdig
6860     }
6861
6862  ⟨∗zpeu | euroitc⟩
6863  \SetProtrusion
6864     { encoding = U,
6865  ⟨zpeu⟩       family   = zpeu,
6866  ⟨euroitc⟩      family   = {euroitc,euroitcs},
6867     shape     = it* }
6868     {
6869  ⟨zpeu⟩       E = {100,-50}
6870  ⟨euroitc⟩      E = {100,}
6871     }
6872
6873  ⟨/zpeu | euroitc⟩
6874  ⟨∗zpeu⟩
6875  \SetProtrusion
6876     { encoding = U,
6877     family    = {zpeus,eurosans} }
6878     {
6879     E = {100,50}
6880     }
6881
6882  \SetProtrusion
6883     { encoding = U,
6884     family    = {zpeus,eurosans},
6885     shape     = it* }
6886     {
6887     E = {200, }
6888     }
6889
6890  ⟨/zpeu⟩
```

---

22   Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

6891 ⟨/cfg − e⟩

15.9   Interword spacing

Default unit is space.

```
6892  ⟨∗m − t⟩
6893  %%% --------------------------------------------------------------------
6894  %%% INTERWORD SPACING
6895
6896  \SetExtraSpacing
6897     [ name = default ]
6898     { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6899     {
```

These settings are only a first approximation. The following reasoning is from a mail from *Ulrich Dirr*. I do not claim to have coped with the task.

'The idea is – analog to the tables for expansion and protrusion – to have tables for optical reduction/expansion of spaces in dependence of the actual character so that the distance between words is optically equal.

When reducing distances the (weighting) order is:

- after commas

```
6900          {,} = { ,-500,500},
```

- in front of capitals which have optical more room on their left side, e. g., 'A', 'J', 'T', 'V', 'W', and 'Y' [this is not yet possible – RS]

- in front of capitals which have circle/oval shapes on their left side, e. g., 'C', 'G', 'O', and 'Q' [ditto – RS]

- after 'r' (because of the bigger optical room on the righthand side)

```
6901          r  = { ,-300,300},
```

- [before or] after lowercase characters with ascenders

```
6902          b  = { ,-200,200},
6903          d  = { ,-200,200},
6904          f  = { ,-200,200},
6905          h  = { ,-200,200},
6906          k  = { ,-200,200},
6907          l  = { ,-200,200},
6908          t  = { ,-200,200},
```

- [before or] after lowercase characters with x-height plus descender with additional optical space, e. g., 'v', or 'w'

```
6909          c  = { ,-100,100},
6910          p  = { ,-100,100},
6911          v  = { ,-100,100},
6912          w  = { ,-100,100},
6913          z  = { ,-100,100},
6914          x  = { ,-100,100},
6915          y  = { ,-100,100},
```

- [before or] after lowercase characters with x-height plus descender without additional optical space

```
6916          i   = { , 50, -50},
6917          m   = { , 50, -50},
6918          n   = { , 50, -50},
6919          u   = { , 50, -50},
```

- after colon and semicolon

```
6920          :   = { ,200,-200},
6921          ;   = { ,200,-200},
```

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

```
6922          .   = { ,250,-250},
6923          !   = { ,250,-250},
6924          ?   = { ,250,-250}
```

The order has to be reversed when enlarging is needed.'

```
6925    }
6926
```

Questions are:

- Is the result really better?

- Is it overdone? (Try with a `factor` < 1000.)

- Should the first parameter also be used? (Probably.)

- What about quotation marks, parentheses etc.?

    Furthermore, there seems to be a pdfTeX bug with spacing in combination with a non-zero `\spaceskip` (reported by *Axel Berger*):

```
\parfillskip0pt
\rightskip0pt plus 1em
\spaceskip\fontdimen2\font
 test test\par
\pdfadjustinterwordglue2
\stbscode\font`t=-50
 test test
\bye
```

Some more characters in T2A.[23]

```
6927  ⟨*m − t⟩
6928  \SetExtraSpacing
6929    [ name      = T2A,
6930      load      = default ]
6931    { encoding = T2A,
6932      family   = cmr }
6933    {
6934      \cyrg  = { ,-300,300},
6935      \cyrb  = { ,-200,200},
6936      \cyrk  = { ,-200,200},
6937      \cyrs  = { ,-100,100},
6938      \cyrr  = { ,-100,100},
6939      \cyrh  = { ,-100,100},
6940      \cyru  = { ,-100,100},
6941      \cyrt  = { , 50, -50},
6942      \cyrp  = { , 50, -50},
```

---

23 Contributed by *Karl Karlsson*.

```
6943       \cyri   = { , 50, -50},
6944       \cyrishrt = { , 50, -50},
6945    }
6946
6947 ⟨/m − t⟩
```

15.9.1 Nonfrenchspacing

The following settings simulate \nonfrenchspacing (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the TEXbook:

'If the space factor f is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if $f \geq 2000$. [. . .] Then the stretch component is multiplied by $f/1000$, while the shrink component is multiplied by $1000/f$.'

The 'extra space' (\fontdimen 7) for Computer Modern Roman is a third of \fontdimen 2, i.e., 333.

```
6948 \SetExtraSpacing
6949    [ name     = nonfrench-cmr,
6950      load     = default,
6951      context  = nonfrench ]
6952    { encoding = {OT1,T1,LY1,OT4,QX,T5},
6953      family   = cmr }
6954    {
```

latex.ltx has:

```
\def\nonfrenchspacing{
  \sfcode`\. 3000
  \sfcode`\? 3000
  \sfcode`\! 3000
```

```
6955      . = {333,2000,-667},
6956      ? = {333,2000,-667},
6957      ! = {333,2000,-667},
```

```
  \sfcode`\: 2000
```

```
6958      : = {333,1000,-500},
```

```
  \sfcode`\; 1500
```

```
6959      ; = {    , 500,-333},
```

```
  \sfcode`\, 1250
```

```
6960    {,}= {    , 250,-200}
```

```
}
```

```
6961    }
6962
```

fontinst, however, which is also used to create the PSNFSS font metrics, sets
\fontdimen 7 to 240 by default. Therefore, the fallback settings use this value for
the first component.

```
6963 \SetExtraSpacing
6964    [ name    = nonfrench-default,
6965      load    = default,
6966      context = nonfrench ]
6967    { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6968    {
6969      . = {240,2000,-667},
6970      ? = {240,2000,-667},
6971      ! = {240,2000,-667},
6972      : = {240,1000,-500},
6973      ; = {    , 500,-333},
6974     {,}= {    , 250,-200}
6975    }
6976
```

15.10 Additional kerning

Default unit is 1 em.

```
6977 %%% -------------------------------------------------------------------------
6978 %%% ADDITIONAL KERNING
6979
```

A dummy list to be loaded when no context is active.

```
6980 \SetExtraKerning
6981    [ name = empty ]
6982    { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
6983    { }
6984
```

15.10.1 French

The ratio of \fontdimen 2 to \fontdimen 6 varies for different fonts, so that either
the kerning of the colon (which should be a space, i. e., \fontdimen 2) or that of the
other punctuation characters (TEX's \thinspace, i. e., one sixth of \fontdimen 6)
may be inaccurate, depending on which unit we choose (space or 1em). For Times,
for example, a thin space would be 665. I don't know whether French typography
really wants a thin space, or rather (as it happens to turn out with CMR) half a
space. (Wikipedia[24] claims it should be a quarter of an em, which seems too much
to me; then again, it also says that this *was* a thin space in French typography.)

```
6985 \SetExtraKerning
6986    [ name    = french-default,
6987      context = french,
6988      unit    = space   ]
6989    { encoding = {OT1,T1,LY1} }
6990    {
6991      :  = {1000,}, % = \fontdimen2
6992      ;  = {500, }, % ~ \thinspace
6993      !  = {500, },
6994      ?  = {500, }
6995    }
```

---

24 http://fr.wikipedia.org/wiki/Espace_typographique, 5 July 2007.

6996

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfTeX.

```
6997 \SetExtraKerning
6998   [ name      = french-guillemets,
6999     context  = french-guillemets,
7000     load     = french-default,
7001     unit     = space   ]
7002   { encoding = {T1,LY1} }
7003   {
7004    \guillemotleft  = { ,800}, % = 0.8\fontdimen2
7005    \guillemotright = {800, }
7006   }
7007
7008 \SetExtraKerning
7009   [ name      = french-guillemets-OT1,
7010     context  = french-guillemets,
7011     load     = french-default,
7012     unit     = space   ]
7013   { encoding = OT1     }
7014   { }
7015
```

15.10.2 Turkish

```
7016 \SetExtraKerning
7017   [ name      = turkish,
7018     context  = turkish ]
7019   { encoding = {OT1,T1,LY1} }
7020   {
7021    :  = {167, }, % = \thinspace
7022    !  = {167, },
7023   {=} = {167, }
7024   }
7025
7026 ⟨/m − t⟩
7027 ⟨/config⟩
```

16 Auxiliary file for micro fine tuning

This file can be used to test protrusion and expansion settings.

```
7028 ⟨∗test⟩
7029 \documentclass{article}
7030
7031 %% Here you can specify the font you want to test, using
7032 %% the commands \fontfamily, \fontseries and \fontshape.
7033 %% Make sure to end all lines with a comment character!
7034 \newcommand*\TestFont{%
7035   \fontfamily{ppl}%
7036 %%  \fontseries{b}%
7037 %%  \fontshape{it}% sc, sl
7038 }
7039
7040 \usepackage{ifthen}
7041 \usepackage[T1]{fontenc}
7042 \usepackage[latin1]{inputenc}
7043 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
```

```
7044
7045 \pagestyle{empty}
7046 \setlength{\parindent}{0pt}
7047 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
7048 \newcommand*\testprotrusion[2][]{%
7049   \ifthenelse{\equal{#1}{r}}{}{#2}%
7050   lorem ipsum dolor sit amet,
7051     \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
7052     \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
7053   you know the rest%
7054   \ifthenelse{\equal{#1}{l}}{}{#2}%
7055   \linebreak
7056   {\fontencoding{\encodingdefault}%
7057   \fontseries{\seriesdefault}%
7058   \fontshape{\shapedefault}%
7059   \selectfont
7060   Here is the beginning of a line, \dotfill and here is its end}\linebreak
7061 }
7062 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
7063 \def\stripprefix#1>{}
7064 \newcount\charcount
7065 \begin{document}
7066
7067 \microtypesetup{expansion=false}
7068
7069 {\centering The font in this document is called by:\\
7070   \texttt{\showTestFont}\par}\bigskip
7071
7072 \TestFont\selectfont
7073  This line intentionally left empty\linebreak
7074 %% A -- Z
7075 \charcount=65
7076 \loop
7077   \testprotrusion{\char\charcount}
7078   \advance\charcount 1
7079   \ifnum\charcount < 91 \repeat
7080 %% a -- z
7081 \charcount=97
7082 \loop
7083   \testprotrusion{\char\charcount}
7084   \advance\charcount 1
7085   \ifnum\charcount < 123 \repeat
7086 %% 0 -- 9
7087 \charcount=48
7088 \loop
7089   \testprotrusion{\char\charcount}
7090   \advance\charcount 1
7091   \ifnum\charcount < 58 \repeat
7092 %%
7093  \testprotrusion[r]{,}
7094  \testprotrusion[r]{.}
7095  \testprotrusion[r]{;}
7096  \testprotrusion[r]{:}
7097  \testprotrusion[r]{?}
7098  \testprotrusion[r]{!}
7099 \testprotrusion[l]{\textexclamdown}
7100 \testprotrusion[l]{\textquestiondown}
7101  \testprotrusion[r]{)}
7102  \testprotrusion[l]{(}
7103  \testprotrusion{/}
```

```
7104  \testprotrusion{\char'\\}
7105  \testprotrusion{-}
7106  \testprotrusion{\textendash}
7107  \testprotrusion{\textemdash}
7108  \testprotrusion{\textquoteleft}
7109  \testprotrusion{\textquoteright}
7110  \testprotrusion{\textquotedblleft}
7111  \testprotrusion{\textquotedblright}
7112  \testprotrusion{\quotesinglbase}
7113  \testprotrusion{\quotedblbase}
7114  \testprotrusion{\guilsinglleft}
7115  \testprotrusion{\guilsinglright}
7116  \testprotrusion{\guillemotleft}
7117  \testprotrusion{\guillemotright}
7118
7119  \newpage
7120  The following displays the current font stretched by 5\%,
7121  normal, and shrunk by 5\%:
7122
7123  \bigskip
7124  \newlength{\MTln}
7125  \newcommand*\teststring
7126    {ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
7127  \settowidth{\MTln}{\teststring}
7128  \microtypesetup{expansion=true}
7129
7130  \parbox{1.05\MTln}{\teststring\linebreak\\
7131                     \teststring}\par\bigskip
7132  \parbox{0.95\MTln}{\teststring}
7133
7134  \end{document}
7135  ⟨/test⟩
```

Needless to say that things may always be improved. For suggestions, mail to
w.m.l@gmx.net.

A The LATEX Project Public License

LPPL Version 1.3c 2008-05-04

Copyright 1999, 2002–2008 LATEX3 Project

Preamble

The LATEX Project Public License (LPPL) is the primary license under which the LATEX kernel and the base LATEX packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This license may be particularly suitable if your work is TEX-related (such as a LATEX package), but it is written in such a way that you can use it even if your work is unrelated to TEX.

The section 'WHETHER AND HOW TO DISTRIBUTE WORKS UNDER THIS LICENSE', below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

This license gives conditions under which a work may be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the LATEX3 Project, believe that the conditions below give you the freedom to make and distribute modified versions of your work that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document 'cfgguide.tex' and 'modguide.tex' in the base LATEX distribution for suggestions.

Definitions

In this license document the following terms are used:

Work: Any work being distributed under this License.

Derived Work: Any work that under any applicable law is derived from the Work.

Modification: Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

Modify: To apply any procedure that produces a Derived Work under any applicable law.

Distribution: Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun's Network File System (NFS).

Compiled Work: A version of the Work that has been processed into a form where it is directly usable on a computer system. This processing may include using installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

Current Maintainer: A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the 'Copyright Holder' under any applicable law.

Base Interpreter: A program or process that is normally needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a 'LATEX-Format' or in the case of files belonging to the 'LATEX-format' a program implementing the 'TEX language'.

Conditions on Distribution and Modification

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.

2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the

Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.

3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.

4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.

5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.

6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.

   (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.

   (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.

   (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless

those persons have stated explicitly that they do provide such support for the Derived Work.

   (d) You distribute at least one of the following with the Derived Work:

      i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;

      ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.

7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.

8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.

9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.

10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.

    (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.

11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.

12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work 'as is', without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any author named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

Maintenance of The Work

The Work has the status 'author-maintained' if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is 'author-maintained'.

The Work has the status 'maintained' if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status 'maintained' to 'unmaintained' if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.

2. If this search is successful, then enquire whether the Work is still maintained.

   (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.

   (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a LATEX work, this could be done, for example, by posting to `comp.text.tex`.)

3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.

   (b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.

4. If you make an 'intention announcement' as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.

5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as 'you' in this section.

Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document 'modguide.tex' in the base LaTeX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing LaTeX under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to LaTeX, the discussion in 'modguide.tex' may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

Here is an example of such a notice and statement:

```
%% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
%    http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status `maintained'.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the 'Work' referring to the three files 'pig.dtx', 'pig.ins', and 'pig.sty' (the last being generated from 'pig.dtx' using 'pig.ins'), the 'Base Interpreter' referring to any 'LaTeX-Format', and both 'Copyright Holder' and 'Current Maintainer' referring to the person 'M. Y. Name'.

If you do not want the Maintenance section of LPPL to apply to your Work, change 'maintained' above into 'author-maintained'. However, we recommend that you use 'maintained' as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

Derived Works That Are Not Replacements

Several clauses of the LPPL specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case (e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

Important Recommendations

Defining What Constitutes the Work
The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee to determine which files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
% This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.