Das microtype Paket

Eine Schnittstelle für die mikrotypographischen Erweiterungen von pdfTEX

R Schlicht ?
w.m.l@gmx.net ?

Zusammenfassung

Das microtype Paket stellt eine LATEX Schnittstelle für die mikrotypographischen Erweiterungen von pdfTEXbereit: 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(≥ 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 XETEX.

Das alternative Paket letterspace, welches ebenso mit einfachem TEXarbeitet, stellt nur die Befehle für Sperrsatz (Letter-spacing) 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. welche es nicht nur zum Werkzeug der Wahl für die Erstellung elektronischer Dokumente, sondern auch Werke überragender, altehrwürdiger Typographie machen: am bedeutendsten *Zeichenvorsprünge (Character Protrusion, auch bekannt als margin kerning)* und*Fontexpansion.* Hàn Thế Thànhs Thesis zitierend:

Nachdem Sie den Text rechterhand gelesen haben, können Sie den Effekt dieser Features betrachten, indem Sie auf die Links klicken:

Protrusion	*off*
Expansion	*off*

Beide Features sind im ganzen Dokument eingeschaltet.

> 'Margin kerning bezeichnet die Justierung der Zeichen an den Rändern (margin) eines gesetzten Textes. Eine vereinfachte Anwendung von Margin kerning ist 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 verfassen ist die Methode, eine weitere oder engere Variante des Fonts zu erschaffen, welche die Wortzwischenräume mehr ausgleichen. 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 schmalern 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 Zeilebreite mittels Fontexpansion zu justieren, sollte bedacht werden, wenn ein Paragraph in Zeilen zerbrochen wird, um bessere Stopppunkte zu wählen.' [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 TEXzu 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 pdfTEXan, 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 angwendet 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 wirds rechts gezeigt, angeführt von einem Sternchen falls es abhängig von der pdfTeX Version und/oder dem Output Modus ist).

3.1 Die mikro-typographischen Features anschalten

`protrusion` `true`, `false`, `compatibility`, `nocompatibility`, ⟨*Name des Font Sets*⟩ *`true`

`expansion` 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).

`activate` Protrusion und Expansion können getrennt (de)aktiviert werden, indem man 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 pdfTEXerstellt):

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

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

```
\usepackage{microtype}
```

Wenn 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 Abschnitt 4 für eine detaillierte Beschreibung. Spezifizieren eines Font Sets für ein Features aktiviert dieses Features implizit mit.

`tracking` `true`, `false`, ⟨*font set name*⟩ `false`

`kerning` `spacing` 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.

3.2 Zeichenvorsprünge

`factor` ⟨*integer*⟩ 1000

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

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

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.

`stretch` `shrink` ⟨*integer*⟩ 20

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 Mengen 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.

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

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 and caveats' Abschnitt (9) nachlesen oder es mit einem größeren `step` probieren.

4 Neuere TEX Systeme benutzen pdfTEX als Standard Engine auch für DVI Ausgaben.

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 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 Changing options later

`\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 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` `\DeclareMicrotypeSet*`

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

Dieses Kommando deklariert ein neues Font Set, auf welches die the mikrotypographischen 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 den NFSS Font Attributen zugewiesene Werte definiert: Kodierung, Familie, Serie, Form und Größe (siehe auch LaTeX 2ε 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 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

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'

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 Character Protrusion

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

Benutzt man dieses Kommando, so kann man die Protrusionsfaktoren 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 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.

6 Für alle Interessierten stellt Tabelle 4 auf Seite 86 die exakte Reihenfolge dar.

Optionen:

`name` Sie können den Protrusionseinstellungen einen Namen zuweisen, sodass Sie in der Lage sind, es 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 Schriftarte erstllen; Einstellungen für andere Formen oder Serien können diese Einstellungen laden, sie erweitern und überschreiben (da der jeweils letzte Wert Vorrang hat). Font Einstellungen 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 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.

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.

5.2 Fontexpansion

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

Standardmäßig können alle Zeichen eines Fonts um denselben Betrag gestrecht 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 58 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 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, komplett in 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 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.

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.

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 Wortzwischenraum, bzw. Streckung und Schrumpfung, welche in Tausendsteln eines 1 em angegeben 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 enttäsuchende 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 Standardeinstellungen stellen Ligaturen nur für das Zeichen '`f`' ab, sprich, 'ff', 'fi', ffi', etc.[10] In außergewöhnlichen Situationen könenn Sie manuell Ligaturen aufbrechen, indem sie

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

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.

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

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 eingegeben wie folgt :

```
\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 eingegeben wie folgt:
\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 pdfTEXzusätzliche 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 × `\fontdimen` 2),ebenso die Schrumpfbeträge der Wortzwischenräume (`\fontdimen` 3 und 4). Umgekehrt würde man, so 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 z.B., 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 Befehle 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}
```

wodurch das Paket – wann immer der Font `lmr` vorkommt und keine Einstellungen dafür gefunden werden können – auch den Font `cmr` ausprobiert. 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, die vom Zusammenhang abhängig sind, zu verwenden. 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 ändern. Zu 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}`

[12] Authoren von Fontpaketen sollten vielleicht einen Blick auf den hook `\Microtype@Hook` werfen,

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 Fussnotenzeichen im Text erhalten, dann können die Einstellungen der Zahlen so definieret 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 verscheidenen 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 Präampel 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*⟩}

---

beschrieben im Implementationsteil, Abschnitt 14.4.3.

Natürlich kennt microtype nicht jede typografische Besonderheit, jeder einzelen 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 zurüchgesetzt. 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 benutzen um kürzere Stücke des Textes zu erzeugen, unabhängig von der Schriftart in dem sich der Schriftsatz befindet.[13] Für solche Ad-hoc-Zeichenabstände, bestizt 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 der 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 für z.B 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 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' ('ſz') niemals getrennt werden, normalerweise sieht man auch 'St.' ('St.') Liganten im letterspacing-Text. Darüber

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.

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 ansich 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’ Liganten werden deaktiviert, innere und äußere Abstände und der äußere Buchstabenabstand wurden auf ihre Standartwerte 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(Bindungen)

\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 Schriebmaschinenschrift 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 Präampel verwendet werden und auch nur einmal. Der Befehl fortert 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 das die spezielle Schriftart, welche Sie gerade verwenden, verschiedene Schriftformen hat, so dass mehr oder weniger Hervorhebung oder Vergrößerungen erforderlich sind. Insbesondere kursive Schriftformen können sich deutlich in unterschiedlichen Schriftarten unterscheiden, deshalb habe Ich mich dageben entscheiden Standarthervorhebungseinstellungen 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-Documente (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 Datein 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 im Inhaltsverzeichnis und ähnliche Listen auf eine Weise verändern, so dass ein Excess-Leader Punkte einfügeführt 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 standartmäßig 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 Präampel 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 Standartsätze 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 zusätzliche 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 zuverstehen 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 mikrotypografische Erweiterung, die mit dem CJK Paket arbeiten kann, die Schriftartvergrößerung.

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, darin 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 MiKTEXin 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 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 beizufügen. 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 Hervorhebungseinstellungen 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 überhautpt nicht erstellt hätte, welches die mikro-typographische Erweiterung einführt und so für die TEX -Welt verfügbar macht. 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 Idee, dass TEX tot ist, und für dass Fixen der Programmfehler die ich gefunden habe.

Harald Harders hat dazu beigetragen die Vorsprungseinstellungen für Adobe Minion möglich zu machen. Ich möchste 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*, das sie das `lua` Modul erschaffen hat.

Ich danke *Philipp Lehman* das er seinem csquotes Paket die Möglichkeit hinzugefügt hat, die ursprüngliche Bedeutung aller aktivierten Zeichen wiederherzustellen, so dass diese für die Zeichen in den Konfigurationsdateien verwendet werden können. Freundlicherweise hat *Peter Wilson* ein Häkchen in seinen ledmac/ledpar Paketen zur Verfügung gestellt, so dass 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 Short history

The comprehensive list of changes can be obtained by running '`makeindex -s gglo.ist -o microtype.gls microtype.glo`'. The following is a list of all changes relevant in the user land; bug and compatibility fixes are swept under the rug. Numbers in brackets indicate the relevant section in this manual.

(?)

- `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 ≥ 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 ≥ 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 $\geq$ 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.

```
\def\MT@MT
⟨package⟩  {microtype}
⟨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.

```
\let\MT@restore@catcodes\@empty
\def\MT@fix@catcode#1#2{%
  \edef\MT@restore@catcodes{%
    \MT@restore@catcodes
    \catcode#1 \the\catcode#1\relax
  }%
  \catcode#1 #2\relax
}
⟨package⟩\MT@fix@catcode{17}{14}% ^^Q (comment)
\MT@fix@catcode{24} {9}% ^^X (ignore)
⟨package⟩\MT@fix@catcode{33}{12}% !
⟨package⟩\MT@fix@catcode{34}{12}% "
\MT@fix@catcode{36} {3}% $ (math shift)
\MT@fix@catcode{39}{12}% '
\MT@fix@catcode{42}{12}% *
\MT@fix@catcode{43}{12}% +
\MT@fix@catcode{44}{12}% ,
\MT@fix@catcode{45}{12}% -
\MT@fix@catcode{58}{12}% :
\MT@fix@catcode{60}{12}% <
\MT@fix@catcode{61}{12}% =
\MT@fix@catcode{62}{12}% >
⟨package⟩\MT@fix@catcode{63}{12}% ?
\MT@fix@catcode{94} {7}% ^ (superscript)
\MT@fix@catcode{96}{12}% `
⟨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.

```
⟨*package⟩
\newcommand*\DeclareMicrotypeSet[3][]{}
\newcommand*\UseMicrotypeSet[2][]{}
\newcommand*\DeclareMicrotypeSetDefault[2][]{}
\newcommand*\SetProtrusion[3][]{}
\newcommand*\SetExpansion[3][]{}
\newcommand*\SetTracking[3][]{}
\newcommand*\SetExtraKerning[3][]{}
\newcommand*\SetExtraSpacing[3][]{}
\newcommand*\DisableLigatures[2][]{}
\newcommand*\DeclareCharacterInheritance[3][]{}
\newcommand*\DeclareMicrotypeVariants[1]{}
\newcommand*\DeclareMicrotypeAlias[2]{}
\newcommand*\LoadMicrotypeFile[1]{}
\newcommand*\DeclareMicrotypeBabelHook[2]{}
\newcommand*\microtypesetup[1]{}
\newcommand*\microtypecontext[1]{}
\newcommand*\textmicrotypecontext[2]{#2}
```

```
\@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
⟨/package⟩
\newcommand*\lsstyle{}
\newcommand\textls[2][]{}
\def\textls#1#{}
\newcommand*\lslig[1]{#1}
⟨*package⟩
}
```

These commands also have a starred version.

```
\def\DeclareMicrotypeSet#1#{\@gobbletwo}
\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.

```
\@onlypreamble\DeclareMicrotypeSet
\@onlypreamble\UseMicrotypeSet
\@onlypreamble\DeclareMicrotypeSetDefault
\@onlypreamble\DisableLigatures
\@onlypreamble\DeclareMicrotypeVariants
\@onlypreamble\DeclareMicrotypeBabelHook
```

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

```
\def\MT@old@cmd#1#2{%
  \newcommand*#1{\MT@warning{%
    \string#1 is deprecated. Please use\MessageBreak
    \string#2 instead}%
    \let #1#2#2}}
\MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
\MT@old@cmd\DeclareMicroTypeSet  \DeclareMicrotypeSet
\MT@old@cmd\UseMicroTypeSet      \UseMicrotypeSet
\MT@old@cmd\LoadMicroTypeFile    \LoadMicrotypeFile
⟨/package⟩
```

`\MT@warning` `\MT@warning@nl` `\MT@info` `\MT@info@nl` `\MT@vinfo` `\MT@error` `\MT@warn@err` Communicate.

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

14.1.1 Debugging

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

0: almost none

1: + sets & lists

2: + heirs

3: + slots

4: + factors

```
⟨*debug⟩
\MT@warning@nl{This is the debug version}
\newcount\tracingmicrotype
\tracingmicrotype=2
\def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
\def\MT@info@nl#1{\PackageInfo\MT@MT{#1\@gobble}\MT@addto@annot{#1}}
\let\MT@vinfo\MT@info@nl
\def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
\def\MT@warning@nl#1{\PackageWarning\MT@MT{#1\@gobble}\MT@addto@annot{Warning: #1}}
\def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
\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

```
\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.

```
\RequirePackage{pdftexcmds}
\newif\ifMT@inannot \MT@inannottrue
\let\MT@pdf@annot\@empty
\def\MT@addto@annot#1{\ifnum\tracingmicrotypeinpdf>\z@ \ifMT@inannot
  {\def\MessageBreak{^^J\@spaces}%
   \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.

```
\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).

```
\def\MT@show@pdfannot#1{%
  \ifnum\tracingmicrotypeinpdf<#1 \else
    \iftracingmicrotypeinpdfall\leavevmode\fi
    \pdfannot height 4pt width 4pt depth 2pt {%
      /Subtype/Caret
      /T(\expandafter\string\font@name)
      \ifcase#1\or
      /Subj(New font)/C[1 0 0]
      \else
      /Subj(Known font)/C[0 1 0]
      \fi
      /Contents(\MT@pdf@annot)
    }%
    \iftracingmicrotypeinpdfall\kern1pt \fi
```

```
    \global\MT@inannotfalse
  \fi
}
⟨/debug⟩
⟨/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`.

```
⟨*plain⟩
\def\MT@plain{2}
\ifx\documentclass\@undefined
  \def\MT@plain{1}
  \def\hmode@bgroup{\leavevmode\bgroup}
  \def\nfss@text#1{{\mbox{#1}}}
  \let\@typeset@protect\relax
  \ifx\eplain\@undefined
    \def\MT@plain{0}
    \def\PackageWarning#1#2{%
      \begingroup
        \newlinechar=10 %
        \def\MessageBreak{^^J(#1)\@spaces\@spaces\@spaces\@spaces}%
        \immediate\write16{^^JPackage #1 Warning: #2\on@line.^^J}%
      \endgroup
    }
    \def\on@line{ on input line \the\inputlineno}
    \def\@spaces{\space\space\space\space}
  \fi
\fi
```

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

```
\def\MT@requires@latex#1{%
  \ifnum\MT@plain<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
}
⟨/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` ($\geq$ 1.30)

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

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

```
\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.

```
\ifx\normalpdftexversion\@undefined \else
  \let\pdftexversion \normalpdftexversion
  \let\pdftexrevision\normalpdftexrevision
  \let\pdfoutput     \normalpdfoutput
\fi
```

Old packages might have let `\pdftexversion` to `\relax`.

```
\ifx\pdftexversion\@undefined \else
  \ifx\pdftexversion\relax \else
⟨debug⟩\MT@dinfo@nl{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
    \def\MT@pdftex@no{7}
⟨*package⟩
    \ifnum\pdftexversion = 140
      \ifnum\pdftexrevision < 4
        \def\MT@pdftex@no{6}
      \fi
    \else
⟨/package⟩
      \ifnum\pdftexversion < 140
        \def\MT@pdftex@no{5}
⟨*package⟩
        \ifnum\pdftexversion < 130
          \def\MT@pdftex@no{4}
          \ifnum\pdftexversion < 120
            \def\MT@pdftex@no{3}
            \ifnum\pdftexversion = 14
              \ifnum \expandafter‘\pdftexrevision < ‘h
                \def\MT@pdftex@no{2}
                \ifnum \expandafter‘\pdftexrevision < ‘f
                  \def\MT@pdftex@no{1}
                \fi
              \fi
            \else
              \ifnum\pdftexversion < 14
                \def\MT@pdftex@no{1}
              \fi
            \fi
          \fi
        \fi
      \fi
⟨/package⟩
    \fi
  \fi
```

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

```
\fi
⟨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.

```
\def\MT@clear@options{%
⟨plain⟩  \MT@requires@latex1{%
  \AtEndOfPackage{\let\@unprocessedoptions\relax}%
  \let\CurrentOption\@empty
⟨plain⟩  }\relax
}

\ifnum\MT@pdftex@no <
⟨package⟩    2
⟨letterspace⟩    6
  \MT@warning@nl{You
    \ifcase\MT@pdftex@no
      don't seem to be using pdftex.\MessageBreak
      `\MT@MT' only works with pdftex.\MessageBreak
      Try running `pdflatex' instead of
        `\ifx\XeTeXversion\@undefined\else xe\fi latex'%
    \else
      are using a pdftex version older than
⟨package⟩      0.14f%
⟨letterspace⟩      1.40%
      .\MessageBreak
      `\MT@MT' does not work with this version.\MessageBreak
      Please install a newer version of pdftex%
    \fi
  }
  \MT@clear@options\MT@restore@catcodes
\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.

```
⟨*!lua | letterspace⟩
\ifx\directlua\@undefined \else
  \ifx\directlua\relax \else
⟨!letterspace⟩    \MT@error
⟨letterspace⟩    \MT@warning@nl
    {`\MT@MT'
⟨!letterspace⟩     only works with luatex if you generate%
⟨letterspace⟩     doesn't currently work with luatex.%
     \MessageBreak
⟨!letterspace⟩     the package with the `lua' option%
⟨letterspace⟩     Bye bye%
    }
⟨!letterspace⟩    {}
⟨letterspace⟩    \MT@clear@options\MT@restore@catcodes
⟨letterspace⟩    \expandafter\expandafter\expandafter\endinput
  \fi
\fi
⟨/!lua | letterspace⟩
```

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

```
\RequirePackage{keyval}[1997/11/10]
⟨*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 ...

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

... and numbers.

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

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

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

\MT@stretch Expansion settings.

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

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

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

```
275 \def\MT@stretch@default{20 }
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
```
⟨/package⟩
\let\MT@letterspace\m@ne
\def\MT@letterspace@default{100}
⟨*package⟩
```

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

```
\newif\ifMT@document
```

14.1.4 Auxiliary macros

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

```
\ifcase 0%
  \ifx\eTeXversion\@undefined 1\else
    \ifx\eTeXversion\relax    1\else
      \ifcase\eTeXversion     1\fi
    \fi
  \fi
\else
  \catcode`\^^Q=9 \catcode`\^^X=14
\fi
⟨debug⟩\MT@dinfo@nl{0}{this is
⟨debug⟩^^Q not
⟨debug⟩  etex}
```

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

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

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

```
⟨*lua⟩
\let\MT@requires@luatex\@secondoftwo
\ifx\directlua\@undefined \else
  \ifx\directlua\relax \else
    \let\MT@requires@luatex\@firstoftwo
  \fi
\fi
⟨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.

```
\MT@requires@luatex{
\ifnum\luatexversion<36
  \def\MT@lua{\directlua0}
\else
  \def\MT@lua{\directlua}
\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.

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

```
  end}
}\relax
⟨/lua⟩
⟨/package⟩
⟨/package | letterspace⟩
```

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

```
⟨*luafile⟩
if microtype then
  -- we simply don't load
else

microtype = {}

microtype.module = {
  name         = "microtype",
  version      = 2.4,
  date         = "2010/01/10",
  description  = "microtype module.",
  author       = "R Schlicht",
  copyright    = "R Schlicht",
  license      = "LPPL",
}

if luatextra and luatextra.provides_module then
  luatextra.provides_module(microtype.module)
end

⟨/luafile⟩
```

To be continued, but first back to primitives.

\MT@glet Here's the forgotten one.

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

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

```
\def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
⟨*package⟩
\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

```
\def\MT@def@n{\MT@exp@cs\def}
\def\MT@gdef@n{\MT@exp@gcs\gdef}
```

\MT@edef@n Its expanding versions.

\MT@xdef@n

```
⟨/package⟩
\def\MT@edef@n{\MT@exp@cs\edef}
⟨*package⟩
\def\MT@xdef@n{\MT@exp@gcs\xdef}
```

\MT@let@nc \let a \csname sequence to a command.

\MT@glet@nc

```
\def\MT@let@nc{\MT@exp@cs\let}
\def\MT@glet@nc{\MT@exp@gcs\MT@glet}
```

\MT@let@cn \let a command to a \csname sequence.

```
\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

```
\def\MT@let@nn{\MT@exp@cs\MT@let@cn}
```

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

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

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

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

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

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

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

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

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

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

`\MT@ifdefined@c@T` `\MT@ifdefined@c@TF` `\MT@ifdefined@n@T` `\MT@ifdefined@n@TF` Wrapper for testing whether command resp. `\csname` sequence is defined. If we are running e-TEX, we will use its primitives `\ifdefined` and `\ifcsname`, which decreases memory use substantially.

```
\def\MT@ifdefined@c@T#1{%
^^X  \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
^^Q  \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
}
⟨/package⟩
\def\MT@ifdefined@c@TF#1{%
^^X  \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
⟨package⟩^^Q  \ifx#1\@undefined
⟨package⟩^^Q    \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
}
\def\MT@ifdefined@n@T#1{%
^^X  \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
⟨package⟩^^Q  \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
⟨package⟩^^Q    \expandafter\@gobble\else\expandafter\@firstofone\fi
}
⟨*package⟩
\def\MT@ifdefined@n@TF#1{%
^^X  \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
^^Q  \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
^^Q    \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
}
```

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

```
\def\MT@detokenize@n#1{%
^^X  \expandafter\MT@rem@last@space\detokenize{#1} \@nil
^^Q  \string#1%
}
\def\MT@detokenize@c#1{%
^^X  \MT@exp@one@n\MT@detokenize@n#1%
^^Q  \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil
}
\def\MT@rem@last@space#1 #2{#1%
  \ifx\@nil#2\else \space
  \expandafter\MT@rem@last@space\expandafter#2\fi
```

```
}
```

`\MT@ifempty` Test whether argument is empty.

```
⟨/package⟩
\begingroup
\catcode`\%=12
\catcode`\&=14
\gdef\MT@ifempty#1{&
  \if %#1%&
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
\endgroup
⟨*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).

```
\MT@requires@pdftex6{
⟨*lua⟩
  \MT@requires@luatex{
    \def\MT@ifint#1{\csname\MT@lua{microtype.ifint([[#1]])}\endcsname}
  }{
⟨/lua⟩
⟨/package⟩
    \def\MT@ifint#1{%
      \ifcase\pdfmatch{^-*[0-9]+ *$}{#1}\relax
        \expandafter\@secondoftwo
      \else
        \expandafter\@firstoftwo
      \fi
    }
⟨*package⟩
⟨lua⟩  }
}{
  \def\MT@ifint#1{%
    \if!\ifnum9<1#1!\else?\fi
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
  }
}
⟨/package⟩
⟨/package | letterspace⟩
⟨*luafile⟩
function microtype.ifint(s)
  if string.find(s,"^-*[0-9]+ *$") then
    tex.write("@firstoftwo")
  else
    tex.write("@secondoftwo")
  end
end

⟨/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.)

```
⟨*package⟩
\MT@requires@pdftex6{
⟨*lua⟩
  \MT@requires@luatex{
    \def\MT@ifdimen#1{\csname\MT@lua{microtype.ifdimen([[#1]])}\endcsname}
  }{
⟨/lua⟩
    \def\MT@ifdimen#1{%
      \ifcase\pdfmatch{^([0-9]+([.,][0-9]+)?|[.,][0-9]+)%
                       (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
        \expandafter\@secondoftwo
      \else
        \expandafter\@firstoftwo
      \fi
    }
⟨lua⟩  }
}{
  \def\MT@ifdimen#1{%
    \setbox\z@=\hbox{%
      \MT@count=1#1\relax
      \ifnum\MT@count=\@ne
        \aftergroup\@secondoftwo
      \else
        \aftergroup\@firstoftwo
      \fi
    }%
  }
}
⟨/package⟩
⟨*luafile⟩
function microtype.ifdimen(s)
  if (string.find(s, "^-*[0-9]+(%a*) *$") or
      string.find(s, "^-*[0-9]*[.,][0-9]+(%a*) *$")) then
    tex.write("@firstoftwo")
  else
    tex.write("@secondoftwo")
  end
end

⟨/luafile⟩
```

`\MT@ifdim` Test floating point numbers.

```
⟨*package⟩
\def\MT@ifdim#1#2#3{%
  \ifdim #1\p@ #2 #3\p@
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
```

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

```
\MT@requires@pdftex5{
⟨*lua⟩
  \MT@requires@luatex{
    \def\MT@ifstreq#1#2{\csname\MT@lua{microtype.ifstreq([[#1]],[[#2]])}\endcsname}
  }{
⟨/lua⟩
    \def\MT@ifstreq#1#2{%
```

```
      \ifcase\pdfstrcmp{#1}{#2}\relax
        \expandafter\@firstoftwo
      \else
        \expandafter\@secondoftwo
      \fi
    }
⟨lua⟩  }
}{
  \def\MT@ifstreq#1#2{%
    \edef\MT@res@a{#1}%
    \edef\MT@res@b{#2}%
    \ifx\MT@res@a\MT@res@b
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
  }
}
⟨/package⟩
⟨*luafile⟩
function microtype.ifstreq(s1, s2)
  if s1 == s2 then
    tex.write("@firstoftwo")
  else
    tex.write("@secondoftwo")
  end
end
```

And here we end the `lua` file.

```
end
⟨/luafile⟩
```

`\MT@xadd` Add item to a list.

```
⟨*package⟩
\def\MT@xadd#1#2{%
  \ifx#1\relax
    \xdef#1{#2}%
  \else
    \xdef#1{#1#2}%
  \fi
}
```

`\MT@xaddb` Add item to the beginning.

```
\def\MT@xaddb#1#2{%
  \ifx#1\relax
    \xdef#1{#2}%
  \else
    \xdef#1{#2#1}%
  \fi
}
⟨/package⟩
```

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

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

```
  \fi
}
\def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
\def\MT@map@clist@#1,{%
  \ifx\@nil#1%
    \expandafter\MT@clist@break
  \fi
  \MT@clist@function{#1}%
  \MT@map@clist@
}
\let\MT@clist@function\@gobble
\def\MT@clist@break#1\@nnil{}
⟨∗package⟩
```

\MT@map@tlist@n
\MT@map@tlist@c
\MT@map@tlist@
\MT@tlist@break

Execute ⟨#2⟩ on all elements of the token list ⟨#1⟩. `\MT@tlist@break` can be used to jump out of the loop.

```
\def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
\def\MT@map@tlist@#1#2{%
  \ifx\@nnil#2\else
    #1{#2}%
    \expandafter\MT@map@tlist@
    \expandafter#1%
  \fi
}
\def\MT@tlist@break#1\@nnil{\fi}
```

\ifMT@inlist@
\MT@in@clist

Test whether item ⟨#1⟩ is in comma list ⟨#2⟩. Using `\pdfmatch` would be slower.

```
\newif\ifMT@inlist@
\def\MT@in@clist#1#2{%
  \def\MT@res@a##1,#1,##2##3\@nnil{%
    \ifx##2\@empty
      \MT@inlist@false
    \else
      \MT@inlist@true
    \fi
  }%
  \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
}
```

\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!

```
\def\MT@rem@from@clist#1#2{%
  \def\MT@res@a##1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
  \def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%
  \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
}
```

\MT@in@tlist
\MT@in@tlist@

Test whether item is in token list. Since this isn't too elegant, I thought that at least here, `\pdfmatch` would be more efficient – however, it turned out to be even slower than this solution.

```
\def\MT@in@tlist#1#2{%
  \MT@inlist@false
  \def\MT@res@a{#1}%
  \MT@map@tlist@c#2\MT@in@tlist@
}
\def\MT@in@tlist@#1{%
  \edef\MT@res@b{#1}%
  \ifx\MT@res@a\MT@res@b
```

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

\MT@in@rlist
\MT@in@rlist@
\MT@in@rlist@@
\MT@size@name

Test whether size \MT@size is in a list of ranges. Store the name of the list in \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
\MT@iterate
\MT@repeat

This is the same as LaTeX's \loop, which we mustn't use, since this could confuse an outer \loop in the document.

```
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
\MT@increment

Increment macro ⟨*#1*⟩ by one. Saves using up too many counters. The e-TeX way 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
```

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

`\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.

```
\def\MT@scale#1#2#3{%
^^Q  \multiply #1 #2\relax
  \ifnum #3 = \z@
^^X    #1=\numexpr #1 * #2\relax
  \else
^^X    #1=\numexpr #1 * #2 / #3\relax
^^Q    \divide #1 #3\relax
  \fi
}
```

`\MT@abbr@pr` `\MT@abbr@ex` Some abbreviations. Thus, we can have short command names but full-length log 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`

```
\def\MT@abbr@pr{protrusion}
\def\MT@abbr@ex{expansion}
\def\MT@abbr@pr@c{protrusion codes}
\def\MT@abbr@ex@c{expansion codes}
\def\MT@abbr@pr@inh{protrusion inheritance}
\def\MT@abbr@ex@inh{expansion inheritance}
\def\MT@abbr@nl{noligatures}
\def\MT@abbr@sp{spacing}
\def\MT@abbr@sp@c{interword spacing codes}
\def\MT@abbr@sp@inh{interword spacing inheritance}
\def\MT@abbr@kn{kerning}
\def\MT@abbr@kn@c{kerning codes}
\def\MT@abbr@kn@inh{kerning inheritance}
\def\MT@abbr@tr{tracking}
\def\MT@abbr@tr@c{tracking amount}
```

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

```
\def\MT@rbba@protrusion{pr}
\def\MT@rbba@expansion{ex}
\def\MT@rbba@spacing{sp}
\def\MT@rbba@kerning{kn}
\def\MT@rbba@tracking{tr}
```

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

```
\def\MT@features{pr,ex,sp,kn,tr}
\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 command 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.

```
\def\MT@is@feature#1{%
  \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
  \ifMT@inlist@
    \expandafter\@firstofone
  \else
    \MT@error{‘\@tempa’ is not an available micro-typographic\MessageBreak
      feature. Ignoring #1}{Available features are: ‘\MT@features@long’.}%
    \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).

```
\MT@requires@pdftex5{
  \def\MT@ledmac@setup{%
    \ifMT@protrusion
      \MT@ifdefined@c@TF\l@dunhbox@line{%
        \MT@info@nl{Patching ledmac to enable character protrusion}%
        \newdimen\MT@led@kern
        \let\MT@led@unhbox@line\l@dunhbox@line
        \renewcommand*{\l@dunhbox@line}[1]{%
          \ifhbox##1%
            \MT@led@kern=\rightmarginkern##1%
            \kern\leftmarginkern##1%
            \MT@led@unhbox@line##1%
            \kern\MT@led@kern
          \fi
        }%
      }{%
        \MT@warning@nl{%
          Character protrusion in paragraphs with line\MessageBreak
          numbering will only work if you update ledmac}%
      }%
    \fi
  }
}{
  \def\MT@ledmac@setup{%
    \ifMT@protrusion
      \MT@warning@nl{%
        The pdftex version you are using does not allow\MessageBreak
        character protrusion in paragraphs with line\MessageBreak
        numbering by the 'ledmac' package.\MessageBreak
        Upgrade pdftex to version 1.30 or later}%
    \fi
  }
}
```

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

```
\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`.)

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

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

```
  \MT@if@false
  \MT@with@babel@and@T{spanish} \MT@if@true
  \MT@with@babel@and@T{galician}\MT@if@true
  \MT@with@babel@and@T{mexican} \MT@if@true
  \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.)

```
  \MT@with@package@T{csquotes}{%
    \@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.

```
  \MT@if@false
  \MT@with@package@T{hyperref}\MT@if@true
  \MT@with@package@T{tex4ht}  \MT@if@true
  \ifMT@if@\MT@restore@p@h\fi
}
```

Check again at the end of the preamble.

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

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

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

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

```
  \let\MT@setupfont@hook\@empty
  \MT@if@false
  \MT@with@babel@and@T{spanish} \MT@if@true
  \MT@with@babel@and@T{galician}\MT@if@true
  \MT@with@babel@and@T{mexican} \MT@if@true
  \ifMT@if@
    \g@addto@macro\MT@setupfont@hook{%
      \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
  \fi
  \MT@with@package@T{csquotes}{%
    \@ifpackagelater{csquotes}{2005/05/11}{%
      \g@addto@macro\MT@setupfont@hook\@disablequotes
    }{%
      \MT@warning@nl{%
        Should you receive warnings about unknown slot\MessageBreak
        numbers, try upgrading the 'csquotes' package}%
    }%
  }%
```

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.

```
  \MT@if@false
⟨/package⟩
⟨plain⟩  \MT@requires@latex2{
  \MT@with@package@T{hyperref}{%
    \pdfstringdefDisableCommands{%
⟨*package⟩
      \let\pickup@font\MT@orig@pickupfont
```

```
      \let\textmicrotypecontext\@secondoftwo
      \let\microtypecontext\@gobble
⟨/package⟩
      \def\lsstyle{\pdfstringdefWarn\lsstyle}%
      \def\textls#1#{\pdfstringdefWarn\textls}%
    }%
⟨package⟩    \MT@if@true
  }%
⟨plain⟩  }\relax
⟨*package⟩
  \MT@with@package@T{tex4ht}\MT@if@true
  \ifMT@if@\g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi
```

The listings package makes numbers and letters active,

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

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

```
    \g@addto@macro\MT@setupfont@hook{%
      \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.

```
      \let\lst@ProcessLetter\@empty
    }%
  }%
```

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.

```
⟨/package⟩
⟨plain⟩  \MT@requires@latex2{
  \MT@with@package@T{soul}{%
    \soulregister\lsstyle 0%
    \soulregister\textls  1%
  }%
```

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

```
⟨*plain⟩
  }{\ifx\SOUL@\@undefined\else
      \soulregister\lsstyle 0%
      \soulregister\textls  1%
    \fi}%
⟨/plain⟩
⟨*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.

```
  \MT@with@package@T{pinyin}{%
    \let\MT@orig@py@macron\py@macron
    \@ifpackagelater{pinyin}{2005/08/11}{% 4.6.0
      \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.

```
  \MT@protrusion
  \MT@expansion
}
```

Interword spacing and kerning (pdfTEX 1.40).

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

Disable ligatures (pdfTEX 1.30).

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

Debugging.

```
⟨debug⟩\MT@show@pdfannot1%
```

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

```
    \MT@register@font
  \fi
}
```

`\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.

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

`\MT@font@copy`

For every new protrusion and expansion contexts, we create a new copy.

```
  \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.

```
  \expandafter\ifx\MT@font@copy\relax
    \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
    \expandafter\ifx\MT@font@orig\relax
      \MT@exp@two@c\MT@glet\MT@font@orig\font@name
    \else
      \MT@exp@two@c\let\font@name\MT@font@orig
    \fi
    \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
⟨debug⟩\MT@dinfo1{creating new copy: \MT@font@copy}%
```

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

```
    \MT@map@clist@c\MT@active@features{%
      \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
        \def\@tempa{##1}%
        \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@rem@from@list
      \fi
    }%
  \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
\MT@feat

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

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

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

```
  \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).

```
    \MT@dotrue
    \edef\@tempa{\csname MT@#1@setname\endcsname}%
    \MT@map@clist@n{font,encoding,family,series,shape,size}{%
      \MT@ifdefined@n@TF{MT@checklist@##1}%
        {\csname MT@checklist@##1\endcsname}%
        {\MT@checklist@{##1}}%
      {#1}%
    }%
  \else
    \MT@dofalse
  \fi
  \ifMT@do
```

\MT@feat stores the current feature.

```
    \def\MT@feat{#1}%
    \csname MT@set@#1@codes\endcsname
  \else
    \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
  \fi
}
```

\MT@dinfo@list

```
⟨debug⟩\def\MT@dinfo@list#1#2#3{\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: #2
⟨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).

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

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

```
    \expandafter\MT@exp@one@n\expandafter\MT@in@clist
      \csname MT@#1\expandafter\endcsname
      \csname MT@#2list@#1@\@tempa\endcsname
    \ifMT@inlist@
⟨debug⟩\MT@dinfo@list{#2}{#1}{in}%
      \MT@dotrue
    \else
⟨debug⟩\MT@dinfo@list{#2}{#1}{not in}%
      \MT@dofalse
      \expandafter\MT@clist@break
    \fi
  }%
```

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.

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

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

```
\def\MT@checklist@family#1{%
⟨!debug⟩  \MT@ifdefined@n@T
⟨debug⟩  \MT@ifdefined@n@TF
      {MT@#1list@family@\@tempa}{%
    \MT@exp@two@n\MT@in@clist
       \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
    \ifMT@inlist@
⟨debug⟩\MT@dinfo@list{#1}{family}{in}%
      \MT@dotrue
    \else
⟨debug⟩\MT@dinfo@list{#1}{family}{not in}%
      \MT@dofalse
      \ifx\MT@familyalias\@empty \else
        \MT@exp@two@n\MT@in@clist
            \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
        \ifMT@inlist@
⟨debug⟩  \MT@dinfo@list{#1}{family alias}{in}%
          \MT@dotrue
⟨debug⟩\else\MT@dinfo@list{#1}{family alias}{not in}%
        \fi
      \fi
    \fi
    \ifMT@do \else
      \expandafter\MT@clist@break
    \fi
  }%
⟨debug⟩  {\MT@dinfo@list{#1}{family}{}}%
}
```

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

```
\def\MT@checklist@size#1{%
⟨!debug⟩  \MT@ifdefined@n@T
⟨debug⟩  \MT@ifdefined@n@TF
      {MT@#1list@size@\@tempa}{%
    \MT@exp@cs\MT@in@rlist{MT@#1list@size@\@tempa}%
    \ifMT@inlist@
⟨debug⟩\MT@dinfo@list{#1}{size}{in}%
      \MT@dotrue
    \else
⟨debug⟩\MT@dinfo@list{#1}{size}{not in}%
      \MT@dofalse
      \expandafter\MT@clist@break
    \fi
  }%
⟨debug⟩  {\MT@dinfo@list{#1}{size}{}}%
}
```

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

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

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

```
    \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
    \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
      \@tempb \csname MT@#1list@font@\@tempa\endcsname
    \ifMT@inlist@
⟨debug⟩\MT@dinfo@list{#1}{font}{in}%
      \expandafter\MT@clist@break
    \else
⟨debug⟩\MT@dinfo@list{#1}{font}{not in}%
      \MT@dofalse
    \fi
  }%
⟨debug⟩  {\MT@dinfo@list{#1}{font}{}}%
}
```

14.2.1 Protrusion

`\MT@protrusion` Set up for protrusion?

```
\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.

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

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

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

Get the name of the inheritance list and parse it.

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

Set an input encoding?

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

Load additional lists?

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

Load the main list.

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

`\MT@get@font@dimen@six` `\MT@dimen@six` If `\fontdimen` 6 is zero, character protrusion, spacing, kerning and tracking won't 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).

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

```
}
```

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

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

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

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

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

```
\def\MT@the@pr@code{\@tempcntb}
\MT@requires@pdftex6{
  \def\MT@the@pr@code@tr{%
    \numexpr\@tempcntb+\MT@letterspace@/2\relax
  }
}\relax
```

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

```
\def\MT@set@codes#1,{%
  \ifx\relax#1\@empty\else
    \MT@split@codes #1==\relax
    \expandafter\MT@set@codes
  \fi
}
```

`\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 different things.

```
\def\MT@split@codes#1=#2=#3\relax{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty \else
    \MT@get@slot
    \ifnum\MT@char > \m@ne
      \MT@get@char@unit
      \csname MT@\MT@feat @split@val\endcsname#2\relax
    \fi
  \fi
}
```

`\MT@pr@split@val`

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

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

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

```
  \MT@ifdefined@c@T\MT@pr@inh@name{%
    \MT@ifdefined@n@T{MT@inh@\MT@pr@inh@name @\MT@char @}{%
      \MT@exp@cs\MT@map@tlist@c
        {MT@inh@\MT@pr@inh@name @\MT@char @}%
        \MT@set@pr@heirs
    }%
  }%
}
```

`\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`.

```
\MT@requires@pdftex3{
  \def\MT@scale@to@em{%
    \@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`.

```
    \MT@scale\@tempcntb \@tempb \MT@dimen@six
    \ifnum\@tempcntb=\z@ \else
      \MT@scale@factor
    \fi
  }
```

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

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

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.

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

```
}{
```

No adjustment with versions 0.14f and 0.14g.

```
  \def\MT@scale@to@em{%
    \MT@count=\@tempb\relax
    \ifnum\MT@count=\z@ \else
      \MT@scale@factor
    \fi
  }
```

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

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

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

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

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

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

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

```
\def\MT@scale@factor{%
  \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
    \expandafter\MT@scale\expandafter \@tempcntb
      \csname MT@\MT@feat @factor@\endcsname \@m
  \fi
  \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
    \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
  \else
    \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
      \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
    \fi
  \fi
}
```

`\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.

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

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

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

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

`\MT@pr@factor@` `\MT@sp@factor@` `\MT@kn@factor@` Apply a factor?

```
  \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
    \MT@let@nn{MT@\MT@feat @factor@}
        {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
    \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
                    \number\csname MT@\MT@feat @factor@\endcsname/1000}%
  }{%
    \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
  }%
```

`\MT@pr@unit@` `\MT@sp@unit@` `\MT@kn@unit@` The `unit` can only be evaluated here, since it might be font-specific. If it's `\@empty`, it's relative to character widths, if it's -1, relative to space dimensions.

```
  \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
    \MT@let@nn{MT@\MT@feat @unit@}%
        {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
    \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
      \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
                      relative to character widths}%
    \else
      \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
        \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
                        relative to width of space}%
      \fi
    \fi
  }{%
    \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
  }%
```

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

```
  \let\MT@get@char@unit\relax
  \let\MT@get@space@unit\@gobble
  \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
    \let\MT@get@char@unit\MT@get@charwd
  \else
    \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
      \let\MT@get@space@unit\MT@get@font@dimen
    \else
      \MT@exp@cs\MT@get@unit{MT@\MT@feat @unit@}%
    \fi
  \fi
```

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

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

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

`\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.

```
\def\MT@get@unit#1{%
  \expandafter\MT@get@unit@#1 e!\@nil
  \ifx\x\@empty\else\let#1\x\fi
  \@defaultunits\@tempdima#1 pt\relax\@nnil
  \ifdim\@tempdima=\z@
    \MT@warning@nl{%
      Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak
      width. Setting factors of list `\@nameuse{MT@\MT@feat @c@name}'\MessageBreak
      relative to character widths instead}%
    \let#1\@empty
    \let\MT@get@char@unit\MT@get@charwd
  \else
    \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} factors relative
                    to \the\@tempdima}%
    \MT@count=\@tempdima\relax
  \fi
}
\def\MT@get@unit@#1e#2#3\@nil{%
  \ifx\\#3\\\let\x\@empty \else
    \if m#2%
      \edef\x{#1\fontdimen6\MT@font}%
    \else
      \if x#2%
        \edef\x{#1\fontdimen5\MT@font}%
      \fi
    \fi
  \fi
}
```

`\MT@set@inputenc` The configurations may be under the regime of an input encoding.

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

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

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

`\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.

```
\MT@addto@setup{%
  \@ifpackageloaded{inputenc}{%
    \@ifpackagelater{inputenc}{2006/02/22}{%
      \def\MT@set@inputenc@{%
        \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
          \MT@load@inputenc
      }%
    }{%
      \let\MT@set@inputenc@\MT@load@inputenc
    }%
  }{%
```

```
    \def\MT@set@inputenc@{%
      \MT@warning@nl{Key ‘inputenc’ used in \MT@curr@list@name, but the ‘inputenc’
          \MessageBreak package isn’t loaded. Ignoring input encoding}%
    }%
  }%
}
```

`\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.

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

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

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

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

```
\def\MT@preset@pr{%
  \expandafter\expandafter\expandafter\MT@preset@pr@
    \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
}
\def\MT@preset@pr@#1,#2\@nil{%
  \ifx\MT@pr@unit@\@empty
    \MT@warn@preset@towidth{pr}%
    \let\MT@preset@aux\MT@preset@aux@factor
  \else
    \def\MT@preset@aux{\MT@preset@aux@space2}%
  \fi
  \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
  \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
  \MT@set@all@pr\@tempa\@tempb
}
```

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

`\MT@preset@aux@factor` `\MT@preset@aux@space`

```
\def\MT@preset@aux@factor#1#2{%
  \@tempcntb=#1\relax
  \MT@scale@factor
  \edef#2{\number\@tempcntb}%
}
\def\MT@preset@aux@space#1#2#3{%
  \def\@tempb{#2}%
  \MT@get@space@unit#1%
  \MT@scale@to@em
  \edef#3{\number\@tempcntb}%
}
```

`\MT@warn@preset@towidth`

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

```
}
```

14.2.2 Expansion

`\MT@expansion` Set up for expansion?

```
\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).

```
\def\MT@set@ex@codes@s{%
  \MT@if@list@exists{%
    \MT@get@ex@opt
    \let\MT@get@char@unit\relax
    \MT@reset@ef@codes
    \MT@get@inh@list
    \MT@set@inputenc{c}%
    \MT@load@list\MT@ex@c@name
    \MT@set@listname
    \MT@let@cn\@tempc{MT@ex@c@\MT@ex@c@name}%
    \expandafter\MT@set@codes\@tempc,\relax,%
    \MT@expandfont
  }\relax
}
```

`\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.

```
\newif\ifMT@nonselected
```

```
\def\MT@set@ex@codes@n{%
  \MT@nonselectedtrue
  \MT@if@list@exists
    \MT@get@ex@opt
  {%
    \let\MT@stretch@  \MT@stretch
    \let\MT@shrink@   \MT@shrink
    \let\MT@step@     \MT@step
    \let\MT@auto@     \MT@auto
    \let\MT@ex@factor@\MT@ex@factor
  }%
  \MT@reset@ef@codes
  \MT@expandfont
  \MT@nonselectedfalse
}
```

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

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

`\MT@expandfont` Expand the font.

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

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

`\MT@reset@ef@codes@`

```
\def\MT@set@all@ex#1{%
⟨debug⟩\MT@dinfo@nl{3}{-- ex: setting all to \number#1}%
  \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
}
\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.

```
\MT@requires@pdftex4{
  \def\MT@reset@ef@codes{%
    \ifnum\MT@ex@factor@=\@m \else
      \MT@reset@ef@codes@
    \fi
  }
}{
  \let\MT@reset@ef@codes\MT@reset@ef@codes@
}
```

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

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

Take an optional factor into account.

```
  \ifnum\MT@ex@factor@=\@m \else
    \MT@scale\@tempcntb \MT@ex@factor@ \@m
  \fi
  \ifnum\@tempcntb > \MT@ex@max
    \MT@warn@ex@too@large\MT@ex@max
  \else
    \ifnum\@tempcntb < \MT@ex@min
      \MT@warn@ex@too@large\MT@ex@min
    \fi
  \fi
  \efcode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%
```

Heirs, heirs, I love thy heirs.

```
  \MT@ifdefined@c@T\MT@ex@inh@name{%
    \MT@ifdefined@n@T{MT@inh@\MT@ex@inh@name @\MT@char @}{%
      \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
    }%
  }%
}
```

`\MT@warn@ex@too@large`

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

`\MT@get@ex@opt` Apply different values to this font?

`\MT@ex@factor@` `\MT@stretch@` `\MT@shrink@` `\MT@step@` `\MT@auto@`

```
\def\MT@get@ex@opt{%
  \MT@set@listname
  \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @factor}{%
    \MT@let@cn\MT@ex@factor@{MT@ex@c@\MT@ex@c@name @factor}%
    \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
  }{%
    \let\MT@ex@factor@\MT@ex@factor
  }%
  \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
```

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

`\MT@get@ex@opt@`

```
\def\MT@get@ex@opt@#1#2{%
  \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @#1}{%
    \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
    \MT@vinfo{... : #2}%
  }{%
    \MT@let@nn{MT@#1@}{MT@#1}%
  }%
}
```

`\MT@set@ex@heirs`

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

`\MT@preset@ex`

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

14.2.3 Interword spacing (glue)

`\MT@spacing` Adjustment of interword spacing?

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

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

```
\def\MT@set@sp@codes{%
  \MT@if@list@exists{%
    \MT@get@font@dimen@six{%
      \MT@get@opt
      \MT@reset@sp@codes
      \MT@get@inh@list
      \MT@set@inputenc{c}%
      \MT@load@list\MT@sp@c@name
      \MT@set@listname
      \MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}%
      \expandafter\MT@set@codes\@tempc,\relax,}%
  }\MT@reset@sp@codes
}
```

`\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).

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

```
    \MT@ifempty\@tempb\relax{%
      \MT@get@space@unit2%
      \MT@scale@to@em
      \knbscode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
    }%
    \def\@tempb{#2}%
    \MT@ifempty\@tempb\relax{%
      \MT@get@space@unit3%
      \MT@scale@to@em
      \stbscode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
    }%
    \def\@tempb{#3}%
    \MT@ifempty\@tempb\relax{%
      \MT@get@space@unit4%
      \MT@scale@to@em
      \shbscode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
    }%
    \MT@ifdefined@c@T\MT@sp@inh@name{%
      \MT@ifdefined@n@T{MT@inh@\MT@sp@inh@name @\MT@char @}{%
        \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@sp@inh@name @\MT@char @}\MT@set@sp@heirs
      }%
    }%
}
```

\MT@set@sp@heirs

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

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

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

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

```
\def\MT@preset@sp{%
  \expandafter\expandafter\expandafter\MT@preset@sp@
    \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
}
\def\MT@preset@sp@#1,#2,#3\@nil{%
  \ifx\MT@sp@unit@\@empty
    \MT@warn@preset@towidth{sp}%
    \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
    \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
    \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
```

```
  \else
    \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
    \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
    \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
  \fi
  \MT@set@all@sp\@tempa\@tempc\@tempb
}
}\relax
```

14.2.4 Additional kerning

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

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

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

```
\def\MT@set@kn@codes{%
  \MT@if@list@exists{%
    \MT@get@font@dimen@six{%
      \MT@get@opt
      \MT@reset@kn@codes
      \MT@get@inh@list
      \MT@set@inputenc{c}%
      \MT@load@list\MT@kn@c@name
      \MT@set@listname
      \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
      \expandafter\MT@set@codes\@tempc,\relax,}%
  }\MT@reset@kn@codes
}
```

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

```
\def\MT@kn@split@val#1,#2\relax{%
  \def\@tempb{#1}%
  \MT@ifempty\@tempb\relax{%
    \MT@get@space@unit2%
    \MT@scale@to@em
    \knbccode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; knbc (\MT@char): \number\knbccode\MT@font\MT@char: [#1]}%
  }%
  \def\@tempb{#2}%
  \MT@ifempty\@tempb\relax{%
    \MT@get@space@unit2%
    \MT@scale@to@em
    \knaccode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
  }%
  \MT@ifdefined@c@T\MT@kn@inh@name{%
    \MT@ifdefined@n@T{MT@inh@\MT@kn@inh@name @\MT@char @}{%
      \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
    }%
  }%
}
```

`\MT@set@kn@heirs`

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

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

`\MT@set@all@kn`
`\MT@reset@kn@codes`
`\MT@reset@kn@codes@`

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

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

```
\def\MT@preset@kn{%
  \expandafter\expandafter\expandafter\MT@preset@kn@
    \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
}
\def\MT@preset@kn@#1,#2\@nil{%
  \ifx\MT@kn@unit@\@empty
    \MT@warn@preset@towidth{kn}%
    \let\MT@preset@aux\MT@preset@aux@factor
  \else
    \def\MT@preset@aux{\MT@preset@aux@space2}%
  \fi
  \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
  \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
  \MT@set@all@kn\@tempa\@tempb
}
}\relax
```

14.2.5 Tracking

This only works with pdfTEX 1.40.

```
\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`

```
\let\MT@tr@font@list\@empty
\def\MT@tracking@{%
  \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
  \ifMT@inlist@\else
    \MT@maybe@do{tr}%
    \ifMT@do\else
      \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
    \fi
  \fi
}
⟨/package⟩
\let\MT@tracking
⟨package⟩  \MT@tracking@
⟨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.

```
\def\MT@set@tr@codes{%
⟨*package⟩
  \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`).

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

Finally, let the letterspaced font propagate.

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

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

```
    \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%
    \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.

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

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

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

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

```
⟨*package⟩
    \else
      \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
                     \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
      \ifdim\MT@outer@kern=\z@\else \MT@ls@outer@k \fi
      \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
                     \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
⟨/package⟩
    \fi
⟨*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).

```
    \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
    \aftergroup\aftergroup\aftergroup\MT@ls@aftergroup
⟨/package⟩
  \fi
⟨package⟩  }%
}
```

`\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@ospace`

```
1688   \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).

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

Now the definitions for the letterspace package with plain TEX.

```
⟨*plain⟩
}{
\def\MT@set@lsfont{\MT@lsfont}
\def\lsstyle{%
  \begingroup
  \escapechar\m@ne
  \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
  \MT@set@tr@codes
  \endgroup
}
\let\textls\@undefined
\let\lslig\@undefined
}
⟨/plain⟩
```

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

```
\DeclareRobustCommand\lslig[1]{%
  {\MT@ifdefined@c@TF\MT@curr@ls{%
     \escapechar\m@ne
     \MT@get@ls@basefont
     \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
     \kern\MT@outer@kern
     \font@name #1%
     \kern\MT@outer@kern%
  }{#1}}%
}
```

`\MT@ls@basefont` `\MT@get@ls@basefont` pdfTEX cannot letterspace fonts that already are letterspaced. Therefore, we have 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.

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

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

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

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

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

```
⟨*package⟩
\MT@requires@pdftex7{
  \def\MT@tr@noligatures{%
    \ifx\MT@tr@ligatures\@empty
      \MT@noligatures@\MT@lsfont\@undefined
    \else
      \MT@noligatures@\MT@lsfont\MT@tr@ligatures
    \fi
  }
}{
  \def\MT@tr@noligatures{%
    \MT@warning@nl{%
      Disabling selected ligatures is only possible since\MessageBreak
      pdftex 1.40.4. Disabling all ligatures instead}%
    \MT@glet\MT@tr@noligatures\relax
  }
}
```

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

```
\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.

```
\def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
⟨debug⟩\MT@dinfo@nl2{... orig. space: \the\fontdimen2\MT@lsfont,
⟨debug⟩    \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
⟨debug⟩    \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
  \let\MT@temp\@empty
  \MT@tr@set@space@{#1}{#4}{2}\@empty
  \MT@tr@set@space@{#2}{#5}{3}\@plus
  \MT@tr@set@space@{#3}{#6}{4}\@minus
  \MT@glet@nc{MT@outer@space\expandafter\string\font@name}\MT@temp
⟨debug⟩\MT@dinfo@nl2{... inner space: \the\fontdimen2\MT@lsfont,
⟨debug⟩    \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
⟨debug⟩\MT@dinfo@nl2{... outer space: \MT@temp}%
}
```

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

```
\def\MT@tr@set@space@#1#2#3#4{%
  \MT@ifempty{#2}{%
    \MT@ifempty{#1}{%
      \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
    }{%
      \MT@tr@set@space@@{#1}{#3}{1000}%
      \edef\MT@temp{\MT@temp#4\the\@tempdima}%
      \fontdimen#3\MT@lsfont=\@tempdima
    }%
  }{%
    \MT@tr@set@space@@{#2}{#3}{2000}%
    \edef\MT@temp{\MT@temp#4\the\@tempdima}%
```

```
    \MT@ifempty{#1}\relax{%
      \MT@tr@set@space@@{#1}{#3}{1000}%
      \fontdimen#3\MT@lsfont=\@tempdima
    }%
  }%
}
```

`\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.

```
\def\MT@tr@set@space@@#1#2#3{%
  \MT@test@ast#1*\@nil{%
    \MT@ifdefined@c@TF\MT@tr@unit@
      {\edef\@tempb{#1}\MT@scale@to@em}
      {\@tempcntb=#1\relax}%
    \@tempdima=\dimexpr \dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
                        -\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).

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

`\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).

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

Disable left outer kerning.

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

The ragged2e package sets `\spaceskip` without glue.

```
        \ifdim\lastskip=%
            \ifnum\spacefactor<2000
              \spaceskip
            \else
              \ifdim\xspaceskip=\z@
                \dimexpr\spaceskip+\fontdimen7\font@name\relax
              \else
                \xspaceskip
              \fi
            \fi
⟨debug⟩\MT@dinfo2{[[[[ adjusting pre space (skip): \the\MT@outer@space}%
          \unskip \hskip\MT@outer@space\relax
          \let\MT@ls@outer@k\relax
        \fi
```

```
      \fi
    \fi
  \fi
}
```

\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).

```
\def\MT@tr@outer@r{%
  \futurelet\MT@tr@outer@next\MT@tr@outer@r@
}
\def\MT@tr@outer@r@{%
  \def\MT@temp*{}%
```

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

```
  \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).

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

        \ifhmode\unkern\fi\egroup
        \MT@set@curr@ok \MT@set@curr@os
        \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\MT@temp=}%
      \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.

```
        \ifx\maybe@ic\MT@tr@outer@next
          \MT@set@curr@ok \MT@set@curr@os
          \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\MT@temp=}%
        \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).

```
          \ifx\check@icr\MT@tr@outer@next
            \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\MT@temp=}%
          \else
            \ifx\@sptoken\MT@tr@outer@next
              \def\MT@temp* {\ifhmode\hskip\MT@outer@space
⟨debug⟩\MT@dinfo2{]]] adjusting post spaces (2): \the\MT@outer@space}%
                \fi}%
            \else
              \ifx~\MT@tr@outer@next
                \def\MT@temp*~{\nobreak\hskip\MT@outer@space
⟨debug⟩\MT@dinfo2{]]] adjusting post spaces (3): \the\MT@outer@space}%
                  }%
              \else
                \ifx\ \MT@tr@outer@next \else
                  \ifx\space\MT@tr@outer@next \else
                    \ifx\@xobeysp\MT@tr@outer@next \else
```

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

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

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

```
\def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
\def\MT@tr@outer@icr@{%
  \let\@let@token= \MT@tr@outer@next
  \maybe@ic@
}
```

For older pdfTEX versions, throw an error.

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

And for luaTEX, too.

```
⟨*lua⟩
\MT@requires@luatex{
  \DeclareRobustCommand\lsstyle{%
    \MT@error{Letterspacing currently doesn't work with luatex}
             {Run pdftex, or use the 'soul' package instead.}%
    \MT@glet\lsstyle\relax
  }
}\relax
⟨/lua⟩
⟨/package⟩
```

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

```
\DeclareRobustCommand\textls{%
  \@ifstar{\let\MT@ls@adjust@\MT@ls@adjust@empty\MT@textls}%
          {\let\MT@ls@adjust@\MT@ls@adjust@relax\MT@textls}%
}
```

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

```
\newcommand\MT@textls[2][]{%
  \ifmmode
    \nfss@text{\MT@ls@set@ls{#1}\lsstyle#2}%
  \else
    \hmode@bgroup
      \MT@ls@set@ls{#1}%
      \lsstyle #2%
      \expandafter
    \egroup
  \fi
}
```

\MT@ls@adjust
\MT@ls@adjust@empty
\MT@ls@adjust@relax
\MT@ls@set@ls

Set current letterspacing amount and outer kerning. This has to be done inside the same group as the letterspacing command.

```
\def\MT@ls@adjust@empty{\let\MT@ls@adjust\@empty}
\def\MT@ls@adjust@relax{\let\MT@ls@adjust\relax}
\def\MT@ls@set@ls#1{%
  \MT@ifempty{#1}%
    {\let\MT@letterspace@\@undefined}%
    {\KV@@sp@def\MT@letterspace@{#1}%
     \MT@ls@too@large\MT@letterspace@}%
  \MT@ls@adjust@
}
```

\MT@ls@too@large

Test whether letterspacing amount is too large.

```
\def\MT@ls@too@large#1{%
  \ifnum#1>\MT@tr@max
    \MT@warning{Maximum for option ‘letterspace’ is \number\MT@tr@max}%
    \let#1\MT@tr@max
  \else
    \ifnum#1<\MT@tr@min
      \MT@warning{Minimum for option ‘letterspace’ is \number\MT@tr@min}%
      \let#1\MT@tr@min
    \fi
  \fi
}
```

\MT@outer@kern
\MT@tr@set@okern

This dimen is used for the starred version of `\textls`, for `\lslig` and for adjusted outer kerning.

```
\newdimen\MT@outer@kern
⟨*package⟩
\def\MT@tr@set@okern#1,#2,{%
  \let\MT@temp\@empty
  \MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
  \MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
  \MT@glet@nc{MT@outer@kern\expandafter\string\font@name}\MT@temp
⟨debug⟩\MT@dinfo@nl2{... outer kerning: (#1,#2)
⟨debug⟩              = \@nameuse{MT@outer@kern\expandafter\string\font@name}}%
}
```

\MT@tr@set@okern@

```
\def\MT@tr@set@okern@#1{%
  \MT@test@ast#1*\@nil{%
    \MT@ifdefined@c@TF\MT@tr@unit@
      {\edef\@tempb{#1}\MT@scale@to@em}
      {\@tempcntb=#1\relax}%
    \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
  }{%
    \MT@ifempty\@tempa{\let\@tempa\@m}\relax
    \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
                        * \fontdimen6\MT@lsfont/2000\relax
  }%
  \advance\@tempdima -\dimexpr \MT@letterspace@ sp
                               * \fontdimen6\MT@lsfont/2000\relax
  \edef\MT@temp{\MT@temp{\the\@tempdima}}%
}
⟨/package⟩
```

\MT@ls@outer@k

Adjust outer kerning.

```
\def\MT@ls@outer@k{\ifhmode\kern\MT@outer@kern\relax\fi}
⟨*package⟩
```

14.2.6 Disabling ligatures

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

```
\MT@requires@pdftex5{
\def\MT@noligatures{%
  \MT@dotrue
  \let\@tempa\MT@nl@setname
  \MT@map@clist@n{font,encoding,family,series,shape,size}{%
    \MT@ifdefined@n@TF{MT@checklist@##1}%
      {\csname MT@checklist@##1\endcsname}%
      {\MT@checklist@{##1}}%
    {nl}%
  }%
  \ifMT@do
    \MT@noligatures@\MT@font\MT@nl@ligatures
  \fi
}
```

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

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

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

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

No '`inputenc`' key.

```
      \let\MT@warn@maybe@inputenc\@empty
      \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
      \MT@map@clist@c#2{%
        \KV@@sp@def\@tempa{##1}\MT@get@slot
        \ifnum\MT@char>\m@ne \tagcode#1\MT@char=\m@ne \fi}%
      \MT@vinfo{... Disabling ligatures for characters: #2}%
    }{%
      \pdfnoligatures#1%
      \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
          know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
          the font instead}%
    }%
  }{%
    \pdfnoligatures#1%
    \MT@vinfo{... Disabling ligatures}%
  }%
}
}\relax
```

14.2.7 Loading the configuration

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

```
\def\MT@load@list#1{%
  \edef\@tempa{#1}%
  \MT@let@cn\@tempb{MT@\MT@feat @c@\@tempa @load}%
  \MT@ifstreq\@tempa\@tempb{%
    \MT@error{\@nameuse{MT@abbr@\MT@feat} list '\@tempa' cannot load itself}{}%
  }{%
    \ifx\@tempb\relax \else
      \MT@ifdefined@n@TF{MT@\MT@feat @c@\@tempb}{%
        \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list '\@tempb'}%
        \begingroup
          \MT@load@list\@tempb
```

```
        \endgroup
        \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list
          \noexpand\MessageBreak`\@tempb'}%
        \MT@let@cn\@tempc{MT@\MT@feat @c@\@tempb}%
        \expandafter\MT@set@codes\@tempc,\relax,%
      }{%
        \MT@error{\@nameuse{MT@abbr@\MT@feat} list `\@tempb' undefined.\MessageBreak
                  Cannot load it from list `\@tempa'}{}%
      }%
    \fi
  }%
}
```

`\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.

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

Check for existence of the file only once.

```
  \MT@in@clist{#1}\MT@file@list
  \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.

```
    \MT@begin@catcodes
      \let\MT@begin@catcodes\relax
      \let\MT@end@catcodes\relax
      \InputIfFileExists{mt-#1.cfg}{%
        \edef\MT@curr@file{mt-#1.cfg}%
        \MT@vinfo{... Loading configuration file \MT@curr@file}%
        \MT@xadd\MT@file@list{#1,}%
      }{%
        \MT@get@basefamily#1\@empty\@empty\@empty\@nil
        \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
        \ifMT@inlist@
          \MT@xadd\MT@file@list{#1,}%
        \else
          \InputIfFileExists{mt-\@tempa.cfg}{%
            \edef\MT@curr@file{mt-\@tempa.cfg}%
            \MT@vinfo{... Loading configuration file \MT@curr@file}%
            \MT@xadd\MT@file@list{\@tempa,#1,}%
          }{%
            \MT@vinfo{... No configuration file mt-#1.cfg}%
            \MT@xadd\MT@file@list{#1,}%
          }%
        \fi
      }%
    \endgroup
  \fi
}
```

`\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 `tabular`s), `!`, `?`, `;`, `:` (`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.

```
\def\MT@cfg@catcodes{%
  \makeatletter
  \catcode`\^7%
  \catcode`\ 9%
  \catcode`\^^I9%
  \catcode`\^^M9%
  \catcode`\\\z@
  \catcode`\{\@ne
  \catcode`\}\tw@
  \catcode`\#6%
  \catcode`\%14%
  \MT@map@tlist@n
    {\!\"\$\&\'\(\)\*\+\,\-\.\/\:\;\<\=\>\?\[\]\_\`\|\~}%
    \@makeother
}
```

`\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.

```
\def\MT@begin@catcodes{%
  \begingroup
  \MT@cfg@catcodes
}
```

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

```
\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.

```
\def\MT@get@basefamily#1#2#3#4\@nil{%
  \ifx\@empty#4%
    \def\@tempa{#1#2#3}%
  \else
    \let\@tempa\@empty
    \edef\@tempb{#1#2#3#4}%
    \expandafter\MT@get@basefamily@\@tempb\@nil
  \fi
}
```

`\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`'.

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

`\MT@listname` `\MT@get@listname` `\MT@get@listname@` Try all combinations of font family, series, shape and size to get a list for the current font.

```
\def\MT@get@listname#1{%
⟨debug⟩\MT@dinfo@nl{1}{trying to find \@nameuse{MT@abbr@#1} list for font `\MT@@font'}%
  \let\MT@listname\@undefined
  \def\@tempb{#1}%
  \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	●	-	●	-	●	-	●	-	●	-	●	-	●	-	●	-

```
}
\def\MT@get@listname@#1{%
  \expandafter\MT@next@listname#1%
  \ifx\MT@listname\@undefined \else
    \expandafter\MT@tlist@break
  \fi
}
```

`\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.

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

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

```
\def\MT@next@listname#1#2#3#4{%
  \edef\@tempa{\MT@encoding
               /\ifnum#1=\@ne \MT@family\fi
               /\ifnum#2=\@ne \MT@series\fi
               /\ifnum#3=\@ne \MT@shape\fi
               /\ifnum#4=\@ne *\fi
                \MT@context}%
⟨debug⟩\MT@dinfo@nl{1}{trying \@tempa}%
  \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
    \MT@next@listname@#4%
  }{%
```

Also try with an alias family.

```
    \ifnum#1=\@ne
      \ifx\MT@familyalias\@empty \else
        \edef\@tempa{\MT@encoding
                     /\MT@familyalias
                     /\ifnum#2=\@ne \MT@series\fi
                     /\ifnum#3=\@ne \MT@shape\fi
                     /\ifnum#4=\@ne *\fi
                      \MT@context}%
⟨debug⟩\MT@dinfo@nl{1}{(alias) \@tempa}%
        \MT@ifdefined@n@T{MT@\@tempb @\@tempa}{%
          \MT@next@listname@#4%
        }%
      \fi
    \fi
  }%
}
```

\MT@next@listname@ If size is to be evaluated, do that, otherwise use the current list.

```
\def\MT@next@listname@#1{%
  \ifnum#1=\@ne
    \MT@exp@cs\MT@in@rlist{MT@\@tempb @\@tempa @sizes}%
    \ifMT@inlist@
      \let\MT@listname\MT@size@name
    \fi
  \else
    \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
  \fi
}
```

\MT@if@list@exists

\MT@context

```
\def\MT@if@list@exists{%
  \MT@let@cn\MT@context{MT@\MT@feat @context}%
  \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
  \MT@get@listname{\MT@feat @c}%
  \MT@ifdefined@c@TF\MT@listname{%
    \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
    \ifMT@nonselected
      \MT@vinfo{... Applying non-selected expansion (list `\MT@listname')}%
    \else
      \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list `\MT@listname'}%
    \fi
    \@firstoftwo
  }{%
```

Since the name cannot be \@empty, this is a sound proof that no matching list exists.

```
    \MT@let@nc{MT@\MT@feat @c@name}\@empty
```

Don't warn if selected=false.

```
    \ifMT@nonselected
      \MT@vinfo{... Applying non-selected expansion (no list)}%
    \else
```

Tracking doesn't require a list, either.

```
      \MT@ifstreq\MT@feat{tr}\relax{%
        \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
          for font\MessageBreak`\MT@@font'%
            \ifx\MT@context\@empty\else\space(context: `\MT@context')\fi.
          Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
      }%
    \fi
    \@secondoftwo
  }%
}
```

\MT@get@inh@list The inheritance lists are global (no context).

\MT@context

```
\def\MT@get@inh@list{%
  \let\MT@context\@empty
  \MT@get@listname{\MT@feat @inh}%
  \MT@ifdefined@c@TF\MT@listname{%
    \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
⟨debug⟩\MT@dinfo@nl{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
⟨debug⟩                `\MT@listname'}%
    \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%
```

If the list is \@empty, it has already been parsed.

```
    \ifx\@tempc\@empty \else
⟨debug⟩\MT@dinfo@nl{1}{parsing inheritance list ...}%
```

The group is only required in case an input encoding is given.

```
      \begingroup
      \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak‘\MT@listname’}%
      \MT@set@inputenc{inh}%
      \expandafter\MT@inh@do\@tempc,\relax,%
      \MT@glet@nc{MT@\MT@feat @inh@\MT@listname}\@empty
      \endgroup
    \fi
  }{%
    \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
  }%
}
```

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@`

```
\def\MT@get@slot{%
  \escapechar‘\\
  \let\MT@char@\m@ne
  \MT@noresttrue
```

Save unexpanded string in case we need to issue a warning message.

```
  \MT@toks=\expandafter{\@tempa}%
```

Now, let's walk through (hopefully) all possible cases.

- It's a letter, a character or a number.

```
  \expandafter\MT@is@letter\@tempa\relax\relax
  \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.

```
    \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.

```
    \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize@c\@tempa}%
      \MT@is@symbol
```

- Now, we'll catch the rest, which hopefully is an accented character (e. g. `\"a`).

```
      {\expandafter\MT@is@composite\@tempa\relax\relax}%
    \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.

```
      \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
          \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
    \fi
  \fi

  \let\MT@char\MT@char@
  \ifnum\MT@char < \z@
    \MT@warn@unknown
  \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.

```
    \ifMT@norest \else
      \MT@warn@rest
      \let\MT@char\m@ne
    \fi
  \fi
  \escapechar\m@ne
}
```

`\ifMT@norest` Test whether all of the string has been used up.

```
\newif\ifMT@norest
```

`\MT@is@letter` Input is a letter, a character or a number.

```
\def\MT@is@letter#1#2\relax{%
  \ifcat a\noexpand#1\relax
    \edef\MT@char@{\number`#1}%
    \ifx\\#2\\%
⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a letter (\MT@char@)}%
    \else
      \MT@norestfalse
    \fi
  \else
    \ifcat !\noexpand#1\relax
      \edef\MT@char@{\number`#1}%
⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a character (\MT@char@)}%
      \ifx\\#2\\%
        \ifnum\MT@char@ > 127 \MT@warn@ascii \fi
      \else
        \MT@norestfalse
        \expandafter\MT@is@number#1#2\relax\relax
      \fi
    \fi
  \fi
}
```

`\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.

```
\def\MT@is@number#1#2#3\relax{%
  \ifx\relax#3\relax \else
```

```
    \ifx\relax#2\relax \else
      \MT@noresttrue
      \if#1"\relax
        \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
⟨debug⟩\MT@dinfo@nl{3}{> ... a hexadecimal number: \MT@char@}%
      \else
        \if#1'\relax
          \def\MT@char@{\number#1#2#3}%
⟨debug⟩\MT@dinfo@nl{3}{> ... an octal number: \MT@char@}%
        \else
          \MT@ifint{#1#2#3}{%
            \def\MT@char@{\number#1#2#3}%
⟨debug⟩\MT@dinfo@nl{3}{> ... a decimal number: \MT@char@}%
          }\MT@norestfalse
        \fi
      \fi
      \ifnum\MT@char@ > \@cclv
        \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
        \let\MT@char@\m@ne
      \fi
    \fi
  \fi
}
```

`\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.

```
\def\MT@is@active#1#2\@nil{%
  \ifnum\catcode`#1 = \active
    \begingroup
      \set@display@protect
      \let\IeC\@firstofone
      \let\@inpenc@undefined@\MT@undefined@char
```

We refrain from checking whether there is a sufficient number of octets.

```
      \def\UTFviii@defined##1{\ifx ##1\relax
        \MT@undefined@char{utf8}\else\expandafter ##1\fi}%
```

For ucs (utf8x). Let's call it experimental ...

```
      \MT@ifdefined@c@T\PrerenderUnicode
        {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
      \edef\x{\endgroup
        \def\noexpand\@tempa{\@tempa}%
```

Append what we think the translation is to the token register we use for the log.

```
        \MT@toks={\the\MT@toks\space(= \@tempa)}%
      }%
    \x
  \fi
}
```

`\MT@undefined@char` For characters not defined in the current input encoding.

```
\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.

```
\def\MT@is@symbol{%
  \expandafter\def\expandafter\MT@char\expandafter
      {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
  \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
      \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
  \ifnum\MT@char@ < \z@
```

. . . or, if it hasn't been defined by `\DeclareTextSymbol`, a letter (e. g., `\i`, when using frenchpro).

```
    \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
  \fi
}
```

`\MT@is@char` A helper macro that inspects the `\meaning` of its argument.

`\MT@charstring`

```
\begingroup
  \catcode`\/=\z@
  /MT@map@tlist@n{/\CHAR}/@makeother
  /lowercase{%
    /def/x{/endgroup
      /def/MT@charstring{\CHAR"}%
      /def/MT@is@char##1\CHAR"##2##3##4/relax{%
        /ifx/relax##1/relax
          /if##3\/relax
            /edef/MT@char@{/number"##2}%
            /MT@ifstreq/MT@charstring{##3##4}/relax/MT@norestfalse
          /else
            /edef/MT@char@{/number"##2##3}%
            /MT@ifstreq/MT@charstring{##4}/relax/MT@norestfalse
          /fi
⟨debug⟩   /MT@dinfo@nl{3}{> ‘/the/MT@toks’ is a \char (/MT@char@)}%
        /fi
      }%
    }%
  }
/x
```

`\MT@is@composite` Here, we are dealing with accented characters, specified as two tokens.

```
\def\MT@is@composite#1#2\relax{%
  \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.

```
    \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
                    \string\csname\MT@encoding\endcsname
                    \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
    \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
```

```
  \fi
}
```

[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 `\mathcode`s and `\mathchardef`s 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`

```
\def\MT@set@listname{%
  \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
    `\@nameuse{MT@\MT@feat @c@name}'}%
}
```

`\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.

```
\def\MT@warn@ascii{%
  \MT@warning@nl{Character `\the\MT@toks' (= \MT@char@)
    is outside of ASCII range.\MessageBreak
    You must load the `inputenc' package before using\MessageBreak
    8-bit characters in \MT@curr@list@name}%
}
```

`\MT@warn@number@too@large` Number too large.

```
\def\MT@warn@number@too@large#1{%
  \MT@warning@nl{%
    Number #1 in encoding `\MT@encoding' too large!\MessageBreak
    Ignoring it in \MT@curr@list@name}%
}
```

`\MT@warn@rest` Not all of the string has been parsed.

```
\def\MT@warn@rest{%
  \MT@warning@nl{%
    Unknown slot number of character\MessageBreak`\the\MT@toks'%
    \MT@warn@maybe@inputenc\MessageBreak
    in font encoding `\MT@encoding'.\MessageBreak
    Make sure it's a single character\MessageBreak
    (or a number) in \MT@curr@list@name}%
}
```

`\MT@warn@unknown` No idea what went wrong.

```
\def\MT@warn@unknown{%
  \MT@warning@nl{%
    Unknown slot number of character\MessageBreak`\the\MT@toks'%
```

```
    \MT@warn@maybe@inputenc\MessageBreak
    in font encoding ‘\MT@encoding’ in \MT@curr@list@name}%
}
```

`\MT@warn@maybe@inputenc` In case an input encoding had been requested.

```
\def\MT@warn@maybe@inputenc{%
  \MT@ifdefined@n@T
    {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}%
    { (input encoding ‘\@nameuse
     {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}’)}%
}
```

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`

```
\let\MT@font@list\@empty
\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.

```
⟨/package⟩
⟨plain⟩\MT@requires@latex2{
\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.

```
  \@ifpackageloaded{CJK}{%
    \@ifpackagelater{CJK}{2006/10/17}% 4.7.0
      {\def\MT@orig@pickupfont{\CJK@ifundefined\CJK@plane}}%
      {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}%
    \g@addto@macro\MT@orig@pickupfont
      {{\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).

```
    \@ifpackageloaded{CJKutf8}%
      {\@ifpackagelater{CJKutf8}{2008/05/22}% 4.8.0
        {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}%
        {\@firstoftwo}}%
      {\@firstoftwo}%
    {\g@addto@macro\MT@orig@pickupfont{%
      {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
         \define@newfont\else\xdef\font@name{%
           \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
    {\g@addto@macro\MT@orig@pickupfont{%
      {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
         \define@newfont\def\CJK@temp{v}%
         \ifx\CJK@temp\CJK@plane
           \expandafter\ifx\csname CJK@cmap@\f@family\CJK@plane\endcsname\relax
           \else\csname CJK@cmap@\f@family\CJK@plane\endcsname\fi
         \else \CJK@addcmap\CJK@plane \fi
       \else\xdef\font@name{%
         \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
  }{%
    \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}%
  }%
```

Check whether \pickup@font is defined as expected. The warning issued by \CheckCommand* would be a bit too generic.

```
  \ifx\pickup@font\MT@orig@pickupfont \else
    \MT@warning@nl{%
      Command \string\pickup@font\space is not defined as expected.%
      \MessageBreak Patching it anyway. Some things may break%
⟨*package⟩
      .\MessageBreak Double-check whether micro-typography is indeed%
      \MessageBreak applied to the document.%
      \MessageBreak (Hint: Turn on 'verbose' mode)%
⟨/package⟩
    }%
  \fi
```

\pickup@font

Then we append our stuff. Everything is done inside a group.

```
  \g@addto@macro\pickup@font{\begingroup}%
```

If the trace package is loaded, we turn off tracing of microtype's setup, which is extremely noisy.

```
  \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
  \g@addto@macro\pickup@font{%
      \escapechar\m@ne
⟨*package⟩
⟨debug⟩       \global\MT@inannottrue
⟨debug⟩       \MT@glet\MT@pdf@annot\@empty
⟨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.

```
      \MT@let@cn\MT@font{MT@subst@\expandafter\string\font@name}%
      \ifx\MT@font\relax
        \let\MT@font\font@name
      \else
        \ifx\MT@font\font@name \else
⟨debug⟩  \MT@addto@annot{= substituted with \MT@@font}%
           \MT@register@subst@font
        \fi
      \fi
      \MT@setupfont
⟨/package⟩
⟨letterspace⟩       \MT@tracking
    \endgroup
  }%
⟨*package⟩
```

`\MT@pickupfont` Remember the patched command for later.

```
  \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.

```
  \g@addto@macro\do@subst@correction
    {\edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
     \MT@glet@nc{MT@subst@\expandafter\string\font@name}\MT@font}%
```

`\add@accent` `\MT@orig@add@accent` Inside `\add@accent`, we have to disable microtype's setup, since the grouping in 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.

```
  \let\MT@orig@add@accent\add@accent
  \def\add@accent#1#2{%
    \let\pickup@font\MT@orig@pickupfont
    \MT@orig@add@accent{#1}{#2}%
    \let\pickup@font\MT@pickupfont
  }%
⟨/package⟩
}
⟨plain⟩}\relax
⟨*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.

```
\def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

`\MT@register@subst@font` Register the substituted font.

```
\def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}
```

`\MT@register@font` Register the current font.

```
\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.

```
\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.

```
\def\MT@check@font@cx{%
  \MT@if@true
  \MT@map@clist@c\MT@active@features{%
    \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
      \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
    \ifMT@inlist@
      \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
    \else
      \MT@if@false
    \fi
  }%
  \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
}
```

`\MT@register@subst@font@cx` Add the substituted font to each feature list.

```
\def\MT@register@subst@font@cx{%
  \MT@map@clist@c\MT@active@features{%
    \MT@exp@cs\MT@xadd
      {MT@##1@\csname MT@##1@context\endcsname font@list}%
      {\font@name,}%
  }%
}
```

`\MT@register@font@cx` For each feature, add the current font to the list, unless we didn't set it up.

```
\def\MT@register@font@cx{%
  \MT@map@clist@c\MT@active@features{%
    \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
      \MT@exp@cs\MT@xadd
        {MT@##1@\csname MT@##1@context\endcsname font@list}%
        {\MT@font,}%
      \def\@tempa{##1}%
      \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
    \fi
  }%
}
```

\MT@maybe@rem@from@list Recurse through all context font lists of the document and remove the font, unless it's the current context.

```
\def\MT@maybe@rem@from@list#1{%
  \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
    \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
       \MT@font \csname MT@\@tempa @#1font@list\endcsname
  }%
}
```

\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.

```
\def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
\MT@addto@setup{%
  \DeclareRobustCommand\microtypecontext[1]{%
    \MT@setup@contexts
    \let\MT@reset@context\relax
    \setkeys{MTC}{#1}%
    \selectfont
    \MT@reset@context
  }%
}
```

\textmicrotypecontext This is just a wrapper around \microtypecontext.

```
\DeclareRobustCommand\textmicrotypecontext[2]{{\microtypecontext{#1}#2}}
```

\MT@reset@context
\MT@reset@context@ We have to reset the font at the end of the group, provided there actually was a change.

```
\def\MT@reset@context@{%
  \MT@vinfo{<<< Resetting contexts\on@line
⟨debug⟩  \MessageBreak= \MT@pr@context/\MT@ex@context
⟨debug⟩                /\MT@tr@context/\MT@kn@context/\MT@sp@context
  }%
  \selectfont
}
```

\MT@setup@contexts The first time \microtypecontext is called, we initialise the context lists and redefine the commands used in \pickup@font.

```
\def\MT@setup@contexts{%
  \MT@map@clist@c\MT@active@features
    {\MT@glet@nc{MT@##1@@font@list}\MT@font@list}%
  \MT@glet\MT@check@font\MT@check@font@cx
  \MT@glet\MT@register@font\MT@register@font@cx
  \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
  \MT@glet\MT@setup@contexts\relax
}
```

Define context keys.

```
\MT@map@clist@c\MT@features@long{%
  \define@key{MTC}{#1}[]{%
    \edef\@tempb{\@nameuse{MT@rbba@#1}}%
    \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
    \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).

```
      \MT@ifempty{##1}{\def\MT@val{@}}{\def\MT@val{##1}}%
```

```
      \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
⟨debug⟩\MT@dinfo{1}{>>> no change of #1 context: `\MT@val'}%
      \else
        \MT@vinfo{>>> Changing #1 context to `\MT@val'\MessageBreak\on@line
⟨debug⟩          \space(previous: `\@nameuse{MT@\@tempb @context}')%
                 }%
        \def\MT@reset@context{\aftergroup\MT@reset@context@}%
```

The next time we see the font, we have to reset *all* factors.

```
        \MT@glet@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes@}%
```

We must also keep track of all contexts in the document.

```
        \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
          \MT@val \csname MT@\@tempb @doc@contexts\endcsname
        \ifMT@inlist@ \else
          \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{{\MT@val}}%
⟨debug⟩  \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
        \fi
        \MT@edef@n{MT@\@tempb @context}{\MT@val}%
      \fi
    \fi
  }%
}
```

\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*

Initialise the contexts.

```
\MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
  \MT@def@n{MT@#1@context}{@}%
  \MT@def@n{MT@#1@doc@contexts}{{@}}%
}
\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`⟨*feature*⟩`list@`⟨*attribute*⟩`@`⟨*set name*⟩. 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.

```
\def\DeclareMicrotypeSet{%
  \@ifstar
    \MT@DeclareSetAndUseIt
    \MT@DeclareSet
}
```

\MT@DeclareSet

```
\newcommand\MT@DeclareSet[3][]{%
  \KV@@sp@def\@tempa{#1}%
  \MT@ifempty\@tempa{%
    \MT@map@clist@c\MT@features{{\MT@declare@sets{##1}{#2}{#3}}}%
  }{%
    \MT@map@clist@c\@tempa{{%
      \KV@@sp@def\@tempa{##1}%
      \MT@ifempty\@tempa\relax{%
        \MT@is@feature{set declaration `#2'}{%
          \MT@exp@one@n\MT@declare@sets
```

```
            {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
        }%
      }%
    }}%
  }%
}
```

`\MT@DeclareSetAndUseIt`

```
\newcommand\MT@DeclareSetAndUseIt[3][]{%
  \MT@DeclareSet[#1]{#2}{#3}%
  \UseMicrotypeSet[#1]{#2}%
}
```

`\MT@curr@set@name` We need to remember the name of the set currently being declared.

```
\let\MT@curr@set@name\@empty
```

`\MT@declare@sets` Define the current set name and parse the keys.

```
\def\MT@declare@sets#1#2#3{%
  \KV@@sp@def\MT@curr@set@name{#2}%
  \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
    \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `\MT@curr@set@name'}%
    \MT@glet@nc{MT@#1list@size@\MT@curr@set@name}\@empty
  }%
  \MT@glet@nc{MT@#1@set@@\MT@curr@set@name}\@empty
⟨debug⟩\MT@dinfo{1}{declaring \@nameuse{MT@abbr@#1} set `\MT@curr@set@name'}%
  \setkeys{MT@#1@set}{#3}%
}
```

`\MT@define@set@key@` ⟨*#1*⟩ = font axis, ⟨*#2*⟩ = feature.

```
\def\MT@define@set@key@#1#2{%
  \define@key{MT@#2@set}{#1}[]{%
    \MT@glet@nc{MT@#2list@#1@\MT@curr@set@name}\@empty
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@get@highlevel{#1}%
```

We do not add the expanded value to the list ...

```
      \MT@exp@two@n\g@addto@macro
        {\csname MT@#2list@#1@\MT@curr@set@name\expandafter\endcsname}%
        {\MT@val,}%
    }%
```

... but keep in mind that the list has to be expanded at the end of the preamble.

```
    \expandafter\g@addto@macro\expandafter\MT@font@sets
      \csname MT@#2list@#1@\MT@curr@set@name\endcsname
⟨debug⟩\MT@dinfo@nl{1}{-- #1: \@nameuse{MT@#2list@#1@\MT@curr@set@name}}%
  }%
}
```

`\MT@get@highlevel` Saying, for instance, '`family=rm*`' or '`shape=bf*`' will expand to `\rmdefault` resp. `\bfdefault`.

```
\def\MT@get@highlevel#1{%
  \expandafter\MT@test@ast\MT@val*\@nil\relax{%
```

And '`family = *`' will become `\familydefault`.

```
    \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
    \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.

```
  }%
}
```

`\MT@test@ast` It the last character is an asterisk, execute the second argument, otherwise the first one.

```
\def\MT@test@ast#1*#2\@nil{%
  \def\@tempa{#1}%
  \MT@ifempty{#2}%
}
```

`\MT@font@sets` Fully expand the font specification and fix catcodes for all font sets.

`\MT@fix@font@set`

```
\let\MT@font@sets\@empty
\def\MT@fix@font@set#1{%
  \xdef#1{#1}%
  \global\@onelevel@sanitize#1%
}
```

`\MT@define@set@key@size` `size` requires special treatment.

```
\def\MT@define@set@key@size#1{%
  \define@key{MT@#1@set}{size}[]{%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \expandafter\MT@get@range\MT@val--\@nil
      \ifx\MT@val\relax \else
        \MT@exp@cs\MT@xadd
          {MT@#1list@size@\MT@curr@set@name}%
          {{{\MT@lower}{\MT@upper}\relax}}%
      \fi
    }%
⟨debug⟩\MT@dinfo@nl{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
  }%
}
```

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`

```
\def\MT@get@range#1-#2-#3\@nil{%
  \MT@ifempty{#1}{%
    \MT@ifempty{#2}{%
      \let\MT@val\relax
    }{%
      \def\MT@lower{0}%
      \def\MT@val{#2}%
      \MT@get@size
      \edef\MT@upper{\MT@val}%
    }%
  }{%
    \def\MT@val{#1}%
    \MT@get@size
    \ifx\MT@val\relax \else
      \edef\MT@lower{\MT@val}%
      \MT@ifempty{#2}{%
        \MT@ifempty{#3}%
          {\def\MT@upper{-1}}%
```

2048 pt is TEX's maximum font size.

```
          {\def\MT@upper{2048}}%
      }{%
        \def\MT@val{#2}%
        \MT@get@size
        \ifx\MT@val\relax \else
          \MT@ifdim\MT@lower>\MT@val{%
            \MT@error{%
              Invalid size range (\MT@lower\space > \MT@val) in font set
              `\MT@curr@set@name'.\MessageBreak Swapping sizes}{}%
            \edef\MT@upper{\MT@lower}%
            \edef\MT@lower{\MT@val}%
          }{%
            \edef\MT@upper{\MT@val}%
          }%
          \MT@ifdim\MT@lower=\MT@upper
            {\def\MT@upper{-1}}%
            \relax
        \fi
      }%
    \fi
  }%
}
```

`\MT@get@size` Translate a size selection command and normalise it.

```
\def\MT@get@size{%
```

A single star would mean `\sizedefault`, which doesn't exist, so we define it to be `\normalsize`.

```
  \if*\MT@val\relax
    \def\@tempa{\normalsize}%
  \else
    \MT@let@cn\@tempa{\MT@val}%
  \fi
  \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).

```
    \begingroup
      \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
      \@tempa\@nil
  \fi
```

Test whether we finally got a number or dimension so that we can strip the 'pt' (`\@defaultunits` and `\strip@pt` are kernel macros).

```
  \MT@ifdimen\MT@val{%
    \@defaultunits\@tempdima\MT@val pt\relax\@nnil
    \edef\MT@val{\strip@pt\@tempdima}%
  }{%
    \MT@warning{Could not parse font size `\MT@val'\MessageBreak
                in font set `\MT@curr@set@name'}%
    \let\MT@val\relax
  }%
}
```

`\MT@define@set@key@font`

```
\def\MT@define@set@key@font#1{%
```

```
  \define@key{MT@#1@set}{font}[]{%
    \MT@glet@nc{MT@#1list@font@\MT@curr@set@name}\@empty
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/*}}\relax
      \expandafter\MT@get@font\MT@val/////\@nil
      \MT@exp@two@n\g@addto@macro
        {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
        {\MT@val,}%
    }%
    \expandafter\g@addto@macro\expandafter\MT@font@sets
      \csname MT@#1list@font@\MT@curr@set@name\endcsname
⟨debug⟩\MT@dinfo@nl{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}%
  }%
}
```

`\MT@get@font` Translate any asterisks.

```
\def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
  \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
  \ifx\MT@val\relax\def\MT@val{0}\fi
  \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
  \let\MT@val\@tempb
}
```

`\MT@get@font@` Helper macro, also used by `\MT@get@font@and@size`.

```
\def\MT@get@font@#1#2#3#4#5#6{%
  \let\@tempb\@empty
  \def\MT@temp{#1/#2/#3/#4/#5}%
  \MT@get@axis{encoding}{#1}%
  \MT@get@axis{family}  {#2}%
  \MT@get@axis{series}  {#3}%
  \MT@get@axis{shape}   {#4}%
  \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
  \MT@ifempty{#5}{%
    \MT@warn@axis@empty{size}{\string\normalsize}%
    \def\MT@val{*}%
  }{%
    \def\MT@val{#5}%
  }%
  \MT@get@size
}
```

`\MT@get@axis`

```
\def\MT@get@axis#1#2{%
  \def\MT@val{#2}%
  \MT@get@highlevel{#1}%
  \MT@ifempty\MT@val{%
    \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
    \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
  }\relax
  \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
}
```

`\MT@warn@axis@empty`

```
\def\MT@warn@axis@empty#1#2{%
  \MT@warning{#1 axis is empty in font specification\MessageBreak
    ‘\MT@temp’. Using ‘#2’ instead}%
}
```

We can finally assemble all pieces to define `\DeclareMicrotypeSet`'s keys. They are also used for `\DisableLigatures`.

```
\MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
  \MT@define@set@key@{encoding}{#1}%
  \MT@define@set@key@{family}  {#1}%
  \MT@define@set@key@{series}  {#1}%
  \MT@define@set@key@{shape}   {#1}%
  \MT@define@set@key@size      {#1}%
  \MT@define@set@key@font      {#1}%
}
```

`\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.

```
\renewcommand*\UseMicrotypeSet[2][]{%
  \KV@@sp@def\@tempa{#1}%
  \MT@ifempty\@tempa{%
    \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
  }{%
    \MT@map@clist@c\@tempa{{%
      \KV@@sp@def\@tempa{##1}%
      \MT@ifempty\@tempa\relax{%
        \MT@is@feature{activation of set `#2'}{%
          \MT@exp@one@n\MT@use@set
            {\csname MT@rbba@\@tempa\endcsname}{#2}%
        }%
      }%
    }}%
  }%
}
```

`\MT@pr@setname` `\MT@ex@setname` `\MT@tr@setname` `\MT@sp@setname` `\MT@kn@setname` `\MT@use@set` Only use sets that have been declared.

```
\def\MT@use@set#1#2{%
  \KV@@sp@def\@tempa{#2}%
  \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
    \MT@xdef@n{MT@#1@setname}{\@tempa}%
  }{%
    \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
      \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
    }%
    \MT@error{%
      The \@nameuse{MT@abbr@#1} set `\@tempa' is undeclared.\MessageBreak
      Using set `\@nameuse{MT@#1@setname}' instead}{}%
  }%
}
```

`\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.

```
\renewcommand*\DeclareMicrotypeSetDefault[2][]{%
  \KV@@sp@def\@tempa{#1}%
  \MT@ifempty\@tempa{%
    \MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
  }{%
    \MT@map@clist@c\@tempa{{%
      \KV@@sp@def\@tempa{##1}%
      \MT@ifempty\@tempa\relax{%
        \MT@is@feature{declaration of default set `#2'}{%
          \MT@exp@one@n\MT@set@default@set
            {\csname MT@rbba@\@tempa\endcsname}{#2}%
        }%
```

```
      }%
    }}%
  }%
}
```

\MT@default@pr@set
\MT@default@ex@set
\MT@default@tr@set
\MT@default@sp@set
\MT@default@kn@set
\MT@set@default@set

```
\def\MT@set@default@set#1#2{%
  \KV@@sp@def\@tempa{#2}%
  \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
⟨debug⟩\MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set ‘\@tempa’}%
    \MT@xdef@n{MT@default@#1@set}{\@tempa}%
  }{%
    \MT@error{%
      The \@nameuse{MT@abbr@#1} set ‘\@tempa’ is not declared.\MessageBreak
      Cannot make it the default set. Using set\MessageBreak ‘all’ instead}{}%
    \MT@xdef@n{MT@default@#1@set}{all}%
  }%
}
```

14.3.2 Variants and aliases

\DeclareMicrotypeVariants
\MT@variants

Specify suffixes for variants (see `fontname/variants.map`). The starred version appends to the list.

```
\let\MT@variants\@empty
\def\DeclareMicrotypeVariants{%
  \@ifstar
    \MT@DeclareVariants
    {\let\MT@variants\@empty\MT@DeclareVariants}%
}
```

\MT@DeclareVariants

```
\def\MT@DeclareVariants#1{%
  \MT@map@clist@n{#1}{%
    \KV@@sp@def\@tempa{##1}%
    \@onelevel@sanitize\@tempa
    \xdef\MT@variants{\MT@variants{\@tempa}}%
  }%
}
```

\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.

```
\renewcommand*\DeclareMicrotypeAlias[2]{%
  \KV@@sp@def\@tempa{#1}%
  \KV@@sp@def\@tempb{#2}%
  \@onelevel@sanitize\@tempb
  \MT@ifdefined@n@T{MT@\@tempa @alias}{%
    \MT@warning{Alias font family ‘\@tempb’ will override
      alias ‘\@nameuse{MT@\@tempa @alias}’\MessageBreak
      for font family ‘\@tempa’}}%
  \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.

```
  \MT@ifdefined@c@T\MT@family{%
⟨debug⟩\MT@dinfo{1}{Activating alias font ‘\@tempb’ for ‘\MT@family’}%
    \MT@glet\MT@familyalias\@tempb
  }%
```

```
}
```

\LoadMicrotypeFile May be used to load a configuration file manually.

```
\def\LoadMicrotypeFile#1{%
  \KV@@sp@def\@tempa{#1}%
  \@onelevel@sanitize\@tempa
  \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
  \ifMT@inlist@
    \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
  \else
    \MT@xadd\MT@file@list{\@tempa,}%
    \MT@begin@catcodes
    \InputIfFileExists{mt-\@tempa.cfg}{%
      \edef\MT@curr@file{mt-\@tempa.cfg}%
      \MT@vinfo{... Loading configuration file \MT@curr@file}%
    }{%
      \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
                      does not exist}%
    }%
    \MT@end@catcodes
  \fi
}
```

14.3.3 Disabling ligatures

\DisableLigatures \MT@DisableLigatures \MT@nl@setname \MT@nl@ligatures This is really simple now: we can re-use the set definitions of \DeclareMicrotypeSet; there can only be one set, which we'll call 'no ligatures'.

The optional argument may be used to disable selected ligatures only.

```
\MT@requires@pdftex5{
\def\DisableLigatures{%
  \MT@begin@catcodes
  \MT@DisableLigatures
}
\newcommand*\MT@DisableLigatures[2][]{%
  \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
  \xdef\MT@active@features{\MT@active@features,nl}%
  \global\MT@noligaturestrue
  \MT@declare@sets{nl}{no ligatures}{#2}%
  \gdef\MT@nl@setname{no ligatures}%
  \MT@end@catcodes
}
}{
```

If pdfTEX is too old, we throw an error.

```
\renewcommand*\DisableLigatures[2][]{%
  \MT@error{Disabling ligatures of a font is only possible\MessageBreak
    with pdftex version 1.30 or newer.\MessageBreak
    Ignoring \string\DisableLigatures}{Upgrade pdftex.}%
}
}
```

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.

```
\def\DeclareMicrotypeBabelHook#1#2{%
  \MT@map@clist@n{#1}{%
```

```
    \KV@@sp@def\@tempa{##1}%
    \MT@gdef@n{MT@babel@\@tempa}{#2}%
  }%
}
```

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).

```
\def\SetProtrusion{%
  \MT@begin@catcodes
  \MT@SetProtrusion
}
```

`\MT@SetProtrusion` We want the catcodes to be correct even if this is called in the preamble.

`\MT@pr@c@name` `\MT@extra@context`

```
\newcommand*\MT@SetProtrusion[3][]{%
  \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.

```
  \MT@set@named@keys{MT@pr@c}{#1}%
⟨debug⟩\MT@dinfo{1}{creating protrusion list `\MT@pr@c@name'}%
  \def\MT@permutelist{pr@c}%
  \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*⟩, ...

```
  \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.

```
  \MT@gdef@n{MT@pr@c@\MT@pr@c@name}{#3}%
  \MT@end@catcodes
}
```

`\SetExpansion` `\SetExpansion` only differs in that it allows some extra options (`stretch`, `shrink`, `step`, `auto`).

```
\def\SetExpansion{%
  \MT@begin@catcodes
  \MT@SetExpansion
}
```

`\MT@SetExpansion`

`\MT@ex@c@name` `\MT@extra@context` `\MT@permutelist`

```
\newcommand*\MT@SetExpansion[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@ex@c}{#1}%
  \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @factor}{%
    \ifnum\csname MT@ex@c@\MT@ex@c@name @factor\endcsname > \@m
      \MT@warning@nl{Expansion factor \number\@nameuse{MT@ex@c@\MT@ex@c@name @factor}
        too large in list\MessageBreak `\MT@ex@c@name'. Setting it to the
        maximum of 1000}%
      \MT@glet@nc{MT@ex@c@\MT@ex@c@name @factor}\@m
    \fi
```

```
  }%
⟨debug⟩\MT@dinfo{1}{creating expansion list ‘\MT@ex@c@name’}%
  \def\MT@permutelist{ex@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \MT@gdef@n{MT@ex@c@\MT@ex@c@name}{#3}%
  \MT@end@catcodes
}
```

\SetTracking

```
\def\SetTracking{%
  \MT@begin@catcodes
  \MT@SetTracking
}
```

\MT@SetTracking Third argument may be empty.

```
\newcommand*\MT@SetTracking[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@tr@c}{#1}%
⟨debug⟩\MT@dinfo{1}{creating tracking list ‘\MT@tr@c@name’}%
  \def\MT@permutelist{tr@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \KV@@sp@def\@tempa{#3}%
  \MT@ifempty\@tempa\relax{%
    \MT@ifint\@tempa
      {\MT@xdef@n{MT@tr@c@\MT@tr@c@name}{\@tempa}}%
      {\MT@warning{Value ‘\@tempa’ is not a number in\MessageBreak
                   tracking set ‘\MT@curr@set@name’}}}%
  \MT@end@catcodes
}
```

\SetExtraSpacing

```
\def\SetExtraSpacing{%
  \MT@begin@catcodes
  \MT@SetExtraSpacing
}
```

\MT@SetExtraSpacing

\MT@sp@c@name
\MT@extra@context
\MT@permutelist

```
\newcommand*\MT@SetExtraSpacing[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@sp@c}{#1}%
⟨debug⟩\MT@dinfo{1}{creating spacing list ‘\MT@sp@c@name’}%
  \def\MT@permutelist{sp@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \MT@gdef@n{MT@sp@c@\MT@sp@c@name}{#3}%
  \MT@end@catcodes
}
```

\SetExtraKerning

```
\def\SetExtraKerning{%
  \MT@begin@catcodes
  \MT@SetExtraKerning
}
```

\MT@SetExtraKerning

\MT@kn@c@name
\MT@extra@context
\MT@permutelist

```
\newcommand*\MT@SetExtraKerning[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@kn@c}{#1}%
⟨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` `\MT@options` We first set the name (if specified), then remove it from the list, and set the 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
```

```
⟨debug⟩                  (context: \MT@extra@context)\fi}%
      \MT@exp@cs\MT@xaddb
        {MT@\MT@permutelist @\@tempb\MT@extra@context @sizes}%
        {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
    }%
  }%
}
```

`\MT@get@font@and@size` Translate any asterisks and split off the size.

```
\def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
  \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
}
```

```
\MT@define@code@key{encoding}{cfg}
\MT@define@code@key{family}  {cfg}
\MT@define@code@key{series}  {cfg}
\MT@define@code@key{shape}   {cfg}
\MT@define@code@key@size     {cfg}
\MT@define@code@key@font     {cfg}
```

`\MT@define@opt@key`

```
\def\MT@define@opt@key#1#2{%
  \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
    \MT@xdef@n{MT@#1@c@\MT@curr@set@name @#2}{##1}}}%
}
```

The options in the optional first argument.

```
\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.

```
  \define@key{MT@#1@c}{name}[]{%
    \MT@ifempty{##1}{%
      \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
    }{%
      \MT@edef@n{MT@#1@c@name}{##1}%
      \MT@ifdefined@n@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%
        \MT@warning{Redefining \@nameuse{MT@abbr@#1} list `\@nameuse{MT@#1@c@name}'}%
      }%
    }%
    \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
  }%
  \MT@define@opt@key{#1}{load}%
  \MT@define@opt@key{#1}{factor}%
  \MT@define@opt@key{#1}{preset}%
  \MT@define@opt@key{#1}{inputenc}%
```

Only one context is allowed. This might change in the future.

```
  \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
}
```

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.

```
\MT@requires@pdftex7{
⟨*lua⟩
  \MT@requires@luatex{
    \define@key{MT@ex@c}{context}[]{%
      \MT@error{Expansion contexts currently don't work with luatex.\MessageBreak
          Ignoring `context' key\on@line}%
```

```
        {Use pdftex instead.}%
    }
  }{
⟨/lua⟩
    \define@key{MT@ex@c}{context}[]{%
      \MT@ifempty{#1}\relax{%
        \MT@glet\MT@copy@font\MT@copy@font@
        \def\MT@extra@context{#1}%
      }%
    }
    \MT@addto@setup{%
      \define@key{MT@ex@c}{context}[]{%
        \ifx\MT@copy@font\MT@copy@font@
          \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
        \else
          \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
             Ignoring 'context' key\on@line}%
            {Either move the settings inside the preamble,\MessageBreak
             or load the package with the 'copyfonts' option.}%
        \fi
      }%
    }
```

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.

```
    \define@key{MT@pr@c}{context}[]{%
      \MT@ifempty{#1}\relax{%
        \MT@glet\MT@copy@font\MT@copy@font@
        \def\MT@extra@context{#1}%
      }%
    }
    \MT@addto@setup{%
      \define@key{MT@pr@c}{context}[]{%
        \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
        \ifx\MT@copy@font\MT@copy@font@\else
          \MT@warning@nl{If protrusion contexts don't work as expected,
            \MessageBreak load the package with the 'copyfonts' option}%
        \fi
      }%
    }
⟨lua⟩  }
}{
  \define@key{MT@ex@c}{context}[]{%
    \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
       or later. Ignoring 'context' key\on@line}%
      {Upgrade pdftex.}%
  }
}
```

`\MT@warn@nodim`

```
\def\MT@warn@nodim#1{%
  \MT@warning{'\@tempa' is not a dimension.\MessageBreak
              Ignoring it and setting values relative to\MessageBreak #1}%
}
```

Protrusion codes may be relative to character width, or to any dimension.

```
\define@key{MT@pr@c}{unit}[character]{%
  \MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}\@empty
  \def\@tempa{#1}%
  \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.

```
    \MT@ifdimen\@tempa
      {\MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}\@tempa}%
      {\MT@warn@nodim{character widths}}%
  }%
}
```

Tracking may only be relative to a dimension.

```
\define@key{MT@tr@c}{unit}[1em]{%
  \MT@glet@nc{MT@tr@c@\MT@curr@set@name @unit}\@empty
  \def\@tempa{#1}%
  \MT@ifdimen\@tempa
    {\MT@glet@nc{MT@tr@c@\MT@curr@set@name @unit}\@tempa}%
    {\MT@warn@nodim{1em}%
     \MT@gdef@n{MT@tr@c@\MT@curr@set@name @unit}{1em}}%
}
```

Spacing and kerning codes may additionally be relative to space dimensions.

```
\MT@map@clist@n{sp,kn}{%
  \define@key{MT@#1@c}{unit}[space]{%
    \MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\@empty
    \def\@tempa{##1}%
    \MT@ifstreq\@tempa{character}\relax{%
      \MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\m@ne
      \MT@ifstreq\@tempa{space}\relax{%
        \MT@ifdimen\@tempa
          {\MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\@tempa}%
          {\MT@warn@nodim{width of space}}%
      }%
    }%
  }%
}
```

The first argument to `\SetExpansion` accepts some more options.

```
\MT@map@clist@n{stretch,shrink,step}{%
  \define@key{MT@ex@c}{#1}[]{%
    \MT@ifempty{##1}\relax{%
      \MT@ifint{##1}{%
```

A space terminates the number.

```
        \MT@gdef@n{MT@ex@c@\MT@curr@set@name @#1}{##1 }%
      }{%
        \MT@warning{%
          Value `##1' for option `#1' is not a number.\MessageBreak
          Ignoring it}%
      }%
    }%
  }%
}
\define@key{MT@ex@c}{auto}[true]{%
  \def\@tempa{#1}%
  \csname if\@tempa\endcsname
```

Don't use `autoexpand` for pdfTeX version older than 1.20.

```
    \MT@requires@pdftex4{%
      \MT@gdef@n{MT@ex@c@\MT@curr@set@name @auto}{autoexpand}%
    }{%
      \MT@warning{pdftex too old for automatic font expansion}%
    }
  \else
    \MT@requires@pdftex4{%
      \MT@glet@nc{MT@ex@c@\MT@curr@set@name @auto}\@empty
    }\relax
  \fi
}
```

Tracking: Interword spacing and outer kerning. The variant with space in case `\SetTracking` is called inside an argument (e. g., to `\IfFileExists`).

```
\MT@define@opt@key{tr}{spacing}
\MT@define@opt@key{tr}{outerspacing}
\MT@define@opt@key{tr}{outerkerning}
```

Which ligatures should be disabled?

```
\define@key{MT@tr@c}{noligatures}[]%
  {\MT@xdef@n{MT@tr@c@\MT@curr@set@name @noligatures}{#1}}
\define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
\define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
\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.

```
\renewcommand*\DeclareCharacterInheritance[1][]{%
  \let\MT@extra@context\@empty
  \let\MT@extra@inputenc\@undefined
  \let\MT@inh@feat\@empty
  \setkeys{MT@inh@}{#1}%
  \MT@begin@catcodes
  \MT@set@inh@list
}
```

`\MT@set@inh@list` Safe category codes.

```
\def\MT@set@inh@list#1#2{%
  \MT@ifempty\MT@inh@feat{%
    \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{#1}{#2}}}%
  }{%
    \MT@map@clist@c\MT@inh@feat{{%
      \KV@@sp@def\@tempa{##1}%
      \MT@ifempty\@tempa\relax{%
        \MT@exp@one@n\MT@declare@char@inh
          {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
      }%
    }}%
```

```
  }%
  \MT@end@catcodes
}
```

The keys for the optional argument.

```
\MT@map@clist@c\MT@features@long{%
  \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}}
\define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}
```

`\MT@declare@char@inh` The lists cannot be given a name by the user.

```
\def\MT@declare@char@inh#1#2#3{%
  \MT@edef@n{MT@#1@inh@name}%
    {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
  \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
  \MT@ifdefined@c@T\MT@extra@inputenc{%
    \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
⟨debug⟩\MT@dinfo{1}{creating inheritance list `\@nameuse{MT@#1@inh@name}'}%
  \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
  \def\MT@permutelist{#1@inh}%
  \setkeys{MT@inh}{#2}%
  \MT@permute
}
```

Parse the second argument. `\DeclareCharacterInheritance` may also be set up for various combinations.

```
\define@key{MT@inh}{encoding}[]{%
  \def\MT@val{#1}%
  \expandafter\MT@encoding@check\MT@val,\@nil
  \MT@get@highlevel{encoding}%
  \MT@edef@n{MT@tempencoding1}{\MT@val}%
}
```

`\MT@encoding@check` But we only allow *one* encoding.

```
\def\MT@encoding@check#1,#2\@nil{%
  \MT@ifempty{#2}\relax{%
    \edef\MT@val{#1}%
    \MT@warning{You may only specify one encoding for character\MessageBreak
                inheritance lists. Ignoring encoding(s) #2}%
  }%
}
```

For the rest, we can reuse the key setup from the configuration lists (`\Set...`).

```
\MT@define@code@key{family}{inh}
\MT@define@code@key{series}{inh}
\MT@define@code@key{shape} {inh}
\MT@define@code@key@size   {inh}
\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 executed only once, namely the first time this inheritance list is encountered (in `\MT@set@`⟨*feature*⟩`@codes`).

```
\def\MT@inh@do#1,{%
  \ifx\relax#1\@empty \else
    \MT@inh@split #1==\relax
    \expandafter\MT@inh@do
  \fi
}
```

\MT@inh@split Only gather the inheriting characters here. Their codes will actually be set in \MT@set@⟨*feature*⟩@codes.

```
\def\MT@inh@split#1=#2=#3\relax{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty \else
    \MT@get@slot
    \ifnum\MT@char > \m@ne
      \let\MT@val\MT@char
      \MT@map@clist@n{#2}{%
        \def\@tempa{##1}%
        \ifx\@tempa\@empty \else
          \MT@get@slot
          \ifnum\MT@char > \m@ne
            \MT@exp@cs\MT@xadd{MT@inh@\MT@listname @\MT@val @}{{\MT@char}}%
          \fi
        \fi
      }%
⟨debug⟩\MT@dinfo@nl{2}{children of #1 (\MT@val):
⟨debug⟩                \@nameuse{MT@inh@\MT@listname @\MT@val @}}%
    \fi
  \fi
}
```

14.3.7 Permutation

\MT@permute \MT@permute@ \MT@permute@@ \MT@permute@@@ \MT@permute@@@@ Calling \MT@permute will define commands for all permutations of the specified font attributes of the form \MT@⟨*list type*⟩@/⟨*encoding*⟩/⟨*family*⟩/⟨*series*⟩/⟨*shape*⟩/⟨/*⟩ to be the expansion of \MT@⟨*list type*⟩@name, i. e., the name of the currently defined list. Size ranges are held in a separate macro called \MT@⟨*list type*⟩@/⟨*font axes*⟩@sizes, which in turn contains the respective ⟨*list name*⟩s attached to the ranges.

```
\def\MT@permute{%
  \let\MT@cnt@encoding\@ne
  \MT@permute@
```

Undefine commands for the next round.

```
  \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
  \MT@glet\MT@tempsize\@undefined
}
\def\MT@permute@{%
  \let\MT@cnt@family\@ne
  \MT@permute@@
  \MT@increment\MT@cnt@encoding
  \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
    \MT@permute@
}
\def\MT@permute@@{%
  \let\MT@cnt@series\@ne
  \MT@permute@@@
  \MT@increment\MT@cnt@family
  \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
    \MT@permute@@
}
\def\MT@permute@@@{%
  \let\MT@cnt@shape\@ne
  \MT@permute@@@@
  \MT@increment\MT@cnt@series
```

```
  \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
    \MT@permute@@@
}
\def\MT@permute@@@@{%
  \MT@permute@@@@@
  \MT@increment\MT@cnt@shape
  \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
    \MT@permute@@@@
}
```

`\MT@permute@@@@@` In order to save some memory, we can ignore unused encodings (inside the document).

```
\def\MT@permute@@@@@{%
  \MT@permute@define{encoding}%
  \ifMT@document
    \ifx\MT@tempencoding\@empty \else
      \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
        {\expandafter\expandafter\expandafter\@gobble}%
    \fi
  \fi
  \MT@permute@@@@@@
}
```

`\MT@permute@@@@@@`

```
\def\MT@permute@@@@@@{%
  \MT@permute@define{family}%
  \MT@permute@define{series}%
  \MT@permute@define{shape}%
  \edef\@tempa{\MT@tempencoding
               /\MT@tempfamily
               /\MT@tempseries
               /\MT@tempshape
               /\MT@ifdefined@c@T\MT@tempsize *}%
```

Some sanity checks: an encoding must be specified (unless nothing else is).

```
  \MT@ifstreq\@tempa{////}\relax{%
    \ifx\MT@tempencoding\@empty
      \MT@warning{%
        You have to specify an encoding for\MessageBreak
        \@nameuse{MT@abbr@\MT@permutelist} list
        `\@nameuse{MT@\MT@permutelist @name}'.\MessageBreak
        Ignoring it}%
    \else
      \MT@ifdefined@c@TF\MT@tempsize{%
```

Add the list of ranges to the beginning of the current combination, after checking for conflicts.

```
        \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}{%
          \MT@map@tlist@c\MT@tempsize\MT@check@rlist
        }%
        \MT@exp@cs\MT@xaddb
          {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
          \MT@tempsize
⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempa,\MessageBreak
⟨debug⟩          sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
⟨debug⟩                          @sizes\endcsname}%
      }{%
```

Only one list can apply to a given combination.

```
        \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context}{%
```

```
          \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
            `\@nameuse{MT@\MT@permutelist @name}' will override list\MessageBreak
            `\@nameuse{MT@\MT@permutelist @\@tempa\MT@extra@context}'
            for font `\@tempa'}%
        }%
⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempa
⟨debug⟩                \ifx\MT@extra@context\@empty\else\MessageBreak
⟨debug⟩                  (context: \MT@extra@context)\fi}%
      }%
      \MT@xdef@n{MT@\MT@permutelist @\@tempa\MT@extra@context}%
          {\csname MT@\MT@permutelist @name\endcsname}%
    \fi
  }%
}
```

`\MT@permute@define` Define the commands.

```
\def\MT@permute@define#1{%
  \@tempcnta=\csname MT@cnt@#1\endcsname\relax
  \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
    {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
    {\MT@let@nc{MT@temp#1}\@empty}%
}
```

`\MT@permute@reset` Reset the commands.

```
\def\MT@permute@reset#1{%
  \@tempcnta=\@ne
  \MT@loop
    \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
    \advance\@tempcnta\@ne
    \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
      \iftrue
      \iffalse
  \MT@repeat
}
```

`\MT@check@rlist` For every new range item in `\MT@tempsize`, check whether it overlaps with ranges in the existing list.

```
\def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}
```

`\MT@check@rlist@` Define the current new range and ...

```
\def\MT@check@rlist@#1#2#3{%
  \def\@tempb{#1}%
  \def\@tempc{#2}%
  \MT@if@false
  \MT@exp@cs\MT@map@tlist@c
    {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
    \MT@check@range
}
```

`\MT@check@range` ... recurse through the list of existing ranges.

```
\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.

```
\def\MT@check@range@#1#2#3{%
  \MT@ifdim{#2}=\m@ne{%
    \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
      \MT@ifdim\@tempb={#1}\MT@if@true\relax
    }{%
```

- Item in list is a simple size, new item is a range.

```
      \MT@ifdim\@tempb>{#1}\relax{%
        \MT@ifdim\@tempc>{#1}{%
          \MT@if@true
          \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
        }\relax
      }%
    }%
  }{%
    \MT@ifdim\@tempc=\m@ne{%
```

- Item in list is a range, new item is a simple size.

```
      \MT@ifdim\@tempb<{#2}{%
        \MT@ifdim\@tempb<{#1}\relax\MT@if@true
      }\relax
    }{%
```

- Both items are ranges.

```
      \MT@ifdim\@tempb<{#2}{%
        \MT@ifdim\@tempc>{#1}{%
          \MT@if@true
          \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
        }\relax
      }\relax
    }%
  }%
  \ifMT@if@
    \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
      `\@nameuse{MT@\MT@permutelist @name}' will override\MessageBreak
      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.

```
    \expandafter\MT@tlist@break
  \fi
}
```

14.4 Package options

14.4.1 Declaring the options

`\ifMT@opt@expansion` `\ifMT@opt@auto` `\ifMT@opt@DVI`

Keep track of whether the user explicitly set these options.

```
\newif\ifMT@opt@expansion
\newif\ifMT@opt@auto
\newif\ifMT@opt@DVI
```

`\MT@optwarn@admissible`

Some warnings.

```
\def\MT@optwarn@admissible#1#2{%
  \MT@warning@nl{`#1' is not an admissible value for option\MessageBreak
                 `#2'. Assuming `false'}%
}
```

`\MT@optwarn@nan`

```
⟨/package⟩
```

```
⟨plain⟩\MT@requires@latex1{
\def\MT@optwarn@nan#1#2{%
  \MT@warning@nl{Value ‘#1’ for option ‘#2’ is not a\MessageBreak number.
                 Using default value of \number\@nameuse{MT@#2@default}}%
}
⟨plain⟩}\relax
⟨∗package⟩
```

`\MT@opt@def@set`

```
\def\MT@opt@def@set#1{%
  \MT@ifdefined@n@TF{MT@\@tempb @set@@\MT@val}{%
    \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
  }{%
    \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
    \MT@warning@nl{The #1 set ‘\MT@val’ is undeclared.\MessageBreak
                   Using set ‘\@nameuse{MT@\@tempb @setname}’ instead}%
  }%
}
```

`expansion` and `protrusion` may be `true`, `false`, `compatibility`, `nocompatibility` and/or a ⟨*set name*⟩.

```
\MT@map@clist@n{protrusion,expansion}{%
  \define@key{MT}{#1}[true]{%
    \csname MT@opt@#1true\endcsname
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifempty\MT@val\relax{%
        \csname MT@#1true\endcsname
        \edef\@tempb{\csname MT@rbba@#1\endcsname}%
        \MT@ifstreq\MT@val{true}\relax
        {%
          \MT@ifstreq\MT@val{false}{%
            \csname MT@#1false\endcsname
          }{%
            \MT@ifstreq\MT@val{compatibility}{%
              \MT@let@nc{MT@\@tempb @level}\@ne
            }{%
              \MT@ifstreq\MT@val{nocompatibility}{%
                \MT@let@nc{MT@\@tempb @level}\tw@
              }{%
```

If everything failed, it should be a set name.

```
                \MT@opt@def@set{#1}%
              }%
            }%
          }%
        }%
      }%
    }%
  }%
}
```

`activate` is a shortcut for `protrusion` and `expansion`.

```
\define@key{MT}{activate}[true]{%
   \setkeys{MT}{protrusion={#1}}%
   \setkeys{MT}{expansion={#1}}%
}
```

`spacing`, `kerning` and `tracking` do not have a compatibility level.

```
\MT@map@clist@n{spacing,kerning,tracking}{%
```

```
  \define@key{MT}{#1}[true]{%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifempty\MT@val\relax{%
        \csname MT@#1true\endcsname
        \MT@ifstreq\MT@val{true}\relax
        {%
          \MT@ifstreq\MT@val{false}{%
            \csname MT@#1false\endcsname
          }{%
            \edef\@tempb{\csname MT@rbba@#1\endcsname}%
            \MT@opt@def@set{#1}%
          }%
        }%
      }%
    }%
  }%
}
```

`\MT@def@bool@opt` The `true`/`false` options: `draft`, `final` (may be inherited from the class options), `auto`, `selected`, `babel`, `DVIoutput`, `defersetup`, `copyfonts`.

```
\def\MT@def@bool@opt#1#2{%
  \define@key{MT}{#1}[true]{%
    \def\@tempa{##1}%
    \MT@ifstreq\@tempa{true}\relax{%
      \MT@ifstreq\@tempa{false}\relax{%
        \MT@optwarn@admissible{##1}{#1}%
        \def\@tempa{false}%
      }%
    }%
    #2%
  }%
}
```

Boolean options that only set the switch.

```
\MT@map@clist@n{draft,selected,babel}{%
  \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
\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.

```
\MT@def@bool@opt{DVIoutput}{%
  \csname if\@tempa\endcsname
    \ifnum\pdfoutput>\z@ \MT@opt@DVItrue \fi
    \pdfoutput\z@
  \else
    \ifnum\pdfoutput<\@ne \MT@opt@DVItrue \fi
    \pdfoutput\@ne
  \fi
}
```

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.

```
\MT@def@bool@opt{defersetup}{%
```

```
  \csname if\@tempa\endcsname \else
    \AtEndOfPackage{%
      \MT@setup@
      \let\MT@setup@\@empty
      \let\MT@addto@setup\@firstofone
    }%
  \fi
}
```

`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.

```
\MT@requires@pdftex7{
⟨*lua⟩
  \MT@requires@luatex{
    \MT@def@bool@opt{copyfonts}{%
      \csname if\@tempa\endcsname
        \MT@error{The 'copyfonts' option doesn't work with luatex}
                 {Use pdftex instead.}%
      \fi
    }
  }{
⟨/lua⟩
    \MT@def@bool@opt{copyfonts}{%
      \csname if\@tempa\endcsname
        \MT@glet\MT@copy@font\MT@copy@font@
      \else
        \MT@glet\MT@copy@font\relax
      \fi
    }
⟨lua⟩  }
}{
  \MT@def@bool@opt{copyfonts}{%
    \csname if\@tempa\endcsname
      \MT@error{The pdftex version you are using is too old\MessageBreak
        to use the 'copyfonts' option}{Upgrade pdftex.}%
    \fi
  }
}
```

`final` is the opposite to `draft`.

```
\MT@def@bool@opt{final}{%
  \csname if\@tempa\endcsname
    \MT@draftfalse
  \else
    \MT@drafttrue
  \fi
}
```

For `verbose` output, we redefine `\MT@vinfo`.

```
\define@key{MT}{verbose}[true]{%
  \let\MT@vinfo\MT@info@nl
  \def\@tempa{#1}%
  \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```
    \MT@ifstreq\@tempa{errors}{%
      \let\MT@warning   \MT@warn@err
      \let\MT@warning@nl\MT@warn@err
    }{%
```

```
      \let\MT@vinfo\@gobble
```

Cast warnings to the winds.

```
      \MT@ifstreq\@tempa{silent}{%
        \let\MT@warning   \MT@info
        \let\MT@warning@nl\MT@info@nl
      }{%
        \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
      }%
    }%
  }%
}
```

Options with numerical keys: `factor`, `stretch`, `shrink`, `step`, `letterspace`.

```
⟨/package⟩
⟨plain⟩\MT@requires@latex1{
\MT@map@clist@n{%
⟨package⟩    stretch,shrink,step,%
    letterspace}{%
  \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
    \def\@tempa{##1 }%
```

No nonsense in `\MT@factor` et al.? A space terminates the number.

```
    \MT@ifint\@tempa
      {\MT@edef@n{MT@#1}{\@tempa}}%
      {\MT@optwarn@nan{##1}{#1}}%
  }%
}
⟨plain⟩}\relax
⟨*package⟩
```

`factor` will define the protrusion factor only.

```
\define@key{MT}{factor}[\MT@factor@default]{%
  \def\@tempa{#1 }%
  \MT@ifint\@tempa
    {\edef\MT@pr@factor{\@tempa}}
    {\MT@optwarn@nan{#1}{factor}}%
}
```

Unit for protrusion codes.

```
\define@key{MT}{unit}[character]{%
  \def\@tempa{#1}%
  \MT@ifstreq\@tempa{character}\relax{%
    \MT@ifdimen\@tempa
      {\let\MT@pr@unit\@tempa}%
      {\MT@warning@nl{`\@tempa' is not a dimension.\MessageBreak
              Ignoring it and setting values relative to\MessageBreak
              character widths}}%
  }%
}
```

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.)

```
\MT@protrusiontrue
\ifnum\pdfoutput<\@ne \else
```

Also, we only enable expansion by default if pdfTEX can expand the fonts automatically.

```
  \MT@requires@pdftex4{
    \MT@expansiontrue
    \MT@autotrue
  }\relax
\fi
```

\MT@config@file
\MT@get@config

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.

```
\define@key{MT}{config}[]{\relax}
\def\MT@get@config#1config=#2,#3\@nil{%
  \MT@ifempty{#2}%
    {\def\MT@config@file{\MT@MT.cfg}}%
    {\def\MT@config@file{#2.cfg}}%
}
\expandafter\expandafter\expandafter\MT@get@config
  \csname opt@\@currname.\@currext\endcsname,config=,\@nil
```

Load the file.

```
\IfFileExists{\MT@config@file}{%
  \MT@info@nl{Loading configuration file \MT@config@file}%
  \MT@begin@catcodes
    \let\MT@begin@catcodes\relax
    \let\MT@end@catcodes\relax
    \let\MT@curr@file\MT@config@file
    \input{\MT@config@file}%
  \endgroup
}{\MT@warning@nl{%
    Could not find configuration file `\MT@config@file'!\MessageBreak
    This will almost certainly cause undesired results.\MessageBreak
    Please fix your installation}%
}
```

\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).

```
\def\MT@check@active@set#1{%
  \MT@ifdefined@n@TF{MT@#1@setname}{%
    \MT@info@nl{Using \@nameuse{MT@abbr@#1} set `\@nameuse{MT@#1@setname}'}%
  }{%
    \MT@ifdefined@n@TF{MT@default@#1@set}{%
      \MT@glet@nn{MT@#1@setname}{MT@default@#1@set}%
      \MT@info@nl{Using default \@nameuse{MT@abbr@#1} set `\@nameuse{MT@#1@setname}'}%
    }{%
```

If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set '`@`', and issue a warning.

```
      \MT@gdef@n{MT@#1@setname}{@}%
      \MT@warning@nl{No \@nameuse{MT@abbr@#1} set chosen, no default set declared.
                     \MessageBreak Using empty set}%
    }%
  }%
}
```

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.

```
\MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
  Command \string\MicroType@Hook\space is deprecated.\MessageBreak
     Use \string\Microtype@Hook\space instead}\MicroType@Hook}
\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.

```
\def\microtypesetup{\setkeys{MT}}
\MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
\def\MT@define@optionX#1#2{%
  \define@key{MTX}{#1}[true]{%
    \edef\@tempb{\csname MT@rbba@#1\endcsname}%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifempty\MT@val\relax{%
        \@tempcnta=\m@ne
        \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.

```
          \MT@checksetup{#1}{%
            \@tempcnta=\csname MT@\@tempb @level\endcsname
            \MT@vinfo{Enabling #1
```

```
                    (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
          }%
        }{%
          \MT@ifstreq\MT@val{false}{%
            \@tempcnta=\z@
            \MT@vinfo{Disabling #1\on@line}%
          }{%
            \MT@ifstreq\MT@val{compatibility}{%
              \MT@checksetup{#1}{%
                \@tempcnta=\@ne
                \MT@let@nc{MT@\@tempb @level}\@ne
                \MT@vinfo{Setting #1 to level 1\on@line}%
              }%
            }{%
              \MT@ifstreq\MT@val{nocompatibility}{%
                \MT@checksetup{#1}{%
                  \@tempcnta=\tw@
                  \MT@let@nc{MT@\@tempb @level}\tw@
                  \MT@vinfo{Setting #1 to level 2\on@line}%
                }%
              }{\MT@error{Value `\MT@val' for key `#1' not recognised}
                         {Use any of `true', `false', `compatibility' or
                          `nocompatibility'.}%
              }%
            }%
          }%
        }%
        \ifnum\@tempcnta>\m@ne
          #2\@tempcnta\relax
        \fi
      }%
    }%
  }%
}
```

`\MT@checksetup` Test whether the feature wasn't disabled in the package options.

```
\def\MT@checksetup#1{%
  \csname ifMT@#1\endcsname
    \expandafter\@firstofone
  \else
    \MT@error{You cannot enable #1 if it was disabled\MessageBreak
              in the package options}{Load microtype with #1 enabled.}%
    \expandafter\@gobble
  \fi
}
```

```
\MT@define@optionX{protrusion}\pdfprotrudechars
\MT@define@optionX{expansion}\pdfadjustspacing
```

`\MT@define@optionX@` The same for `tracking`, `spacing` and `kerning`, which do not have a `compatibility` level.

```
\MT@requires@pdftex6{
⟨lua⟩  \MT@requires@luatex\@firstofone{
  \def\MT@define@optionX@#1#2{%
    \define@key{MTX}{#1}[true]{%
      \MT@map@clist@n{##1}{%
        \KV@@sp@def\MT@val{####1}%
        \MT@ifempty\MT@val\relax{%
          \@tempcnta=\m@ne
          \MT@ifstreq\MT@val{true}{%
```

```
            \MT@checksetup{#1}{%
              \@tempcnta=\@ne
              \MT@vinfo{Enabling #1\on@line}%
            }%
          }{%
            \MT@ifstreq\MT@val{false}{%
              \@tempcnta=\z@
              \MT@vinfo{Disabling #1\on@line}%
            }{\MT@error{Value `\MT@val' for key `#1' not recognised}
                       {Use either `true' or `false'}%
            }%
          }%
          \ifnum\@tempcnta>\m@ne
            #2\relax
          \fi
        }%
      }%
    }%
  }
```

We cannot simply let `\MT@tracking` relax, since this may select the already letterspaced font instance.

```
  \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
                                \else \let\MT@tracking\MT@tracking@ \fi}
  \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
  \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
                               \pdfappendkern \@tempcnta}
  \@gobble
⟨lua⟩  }
}\@firstofone
```

Disable for older pdfTEX versions and for luaTEX.

```
{\define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
}
\define@key{MTX}{activate}[true]{%
  \setkeys{MTX}{protrusion={#1}}%
  \setkeys{MTX}{expansion={#1}}%
}
```

`\MT@saved@setupfont` Disable everything – may be used as a work-around in case setting up fonts doesn't work in certain environments. *(Undocumented.)*

```
\let\MT@saved@setupfont\MT@setupfont
```

```
\define@key{MTX}{disable}[]{%
  \MT@info{Inactivate `\MT@MT' package}%
  \let\MT@setupfont\relax
}
\define@key{MTX}{enable}[]{%
  \MT@info{Reactivate `\MT@MT' package}%
  \let\MT@setupfont\MT@saved@setupfont
}
⟨/package⟩
```

14.4.5 Processing the options

`\MT@ProcessOptionsWithKV` Parse options.

```
⟨plain⟩\MT@requires@latex1{
```

```
\def\MT@ProcessOptionsWithKV#1{%
  \let\@tempc\relax
  \let\MT@temp\@empty
⟨plain⟩  \MT@requires@latex2{
    \MT@map@clist@c\@classoptionslist{%
      \def\CurrentOption{##1}%
      \MT@ifdefined@n@T{KV@#1@\expandafter\MT@getkey\CurrentOption=\@nil}{%
        \edef\MT@temp{\MT@temp,\CurrentOption,}%
        \@expandtwoargs\@removeelement\CurrentOption
          \@unusedoptionlist\@unusedoptionlist
      }%
    }%
    \edef\MT@temp{\noexpand\setkeys{#1}%
                    {\MT@temp\@ptionlist{\@currname.\@currext}}}%
```

eplain can handle package options.

```
⟨*plain⟩
  }{\edef\MT@temp{\noexpand\setkeys{#1}%
                    {\csname usepkg@options@\usepkg@pkg\endcsname}}}
⟨/plain⟩
  \MT@temp
  \MT@clear@options
}
```

`\MT@getkey` For `key=val` in class options.

```
\def\MT@getkey#1=#2\@nil{#1}

\MT@ProcessOptionsWithKV{MT}
⟨plain⟩}\relax
⟨*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).

```
\MT@addto@setup{%
\ifMT@draft
```

We disable most of what we've just defined in the 3748 lines above if we are running in draft mode.

```
  \MT@warning@nl{‘draft’ option active.\MessageBreak
                 Disabling all micro-typographic extensions.\MessageBreak
                 This might lead to different line and page breaks}%
  \let\MT@setupfont\relax
  \renewcommand*\LoadMicrotypeFile[1]{}%
  \renewcommand*\microtypesetup[1]{}%
  \renewcommand*\microtypecontext[1]{}%
  \renewcommand*\lsstyle{}%
\else
```

For DVI output, the user must have explicitly passed the `expansion` option to the package.

```
  \ifnum\pdfoutput<\@ne
    \ifMT@opt@expansion \else
      \MT@expansionfalse
    \fi
  \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!

```
\MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
            \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
```

Working on font copies?

```
\ifx\MT@copy@font\relax\else \MT@info@nl{Using font copies for contexts}\fi
```

Fix the font sets.

```
\MT@map@tlist@c\MT@font@sets\MT@fix@font@set
```

Protrusion.

```
\ifMT@protrusion
  \edef\MT@active@features{\MT@active@features,pr}%
  \pdfprotrudechars\MT@pr@level
  \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
    \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
      factor: \number\MT@pr@factor\fi
    \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}%
  \MT@check@active@set{pr}%
\else
  \let\MT@protrusion\relax
  \MT@info@nl{No character protrusion}%
\fi
```

Expansion.

```
\ifMT@expansion
```

Set up the values for font expansion: if `stretch` has not been specified, we take the default value of 20.

```
  \ifnum\MT@stretch=\m@ne
    \let\MT@stretch\MT@stretch@default
  \fi
```

If `shrink` has not been specified, it will inherit the value from `stretch`.

```
  \ifnum\MT@shrink=\m@ne
    \let\MT@shrink\MT@stretch
  \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.

```
  \MT@requires@pdftex6{\def\MT@step{1 }}{%
    \ifnum\MT@step=\m@ne
      \ifnum\MT@stretch>\MT@shrink
        \ifnum\MT@shrink=\z@
          \@tempcnta=\MT@stretch
        \else
          \@tempcnta=\MT@shrink
        \fi
      \else
        \ifnum\MT@stretch=\z@
          \@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
```

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

Filter out `stretch=0,shrink=0`, since it would result in a pdfTEX error.

```
    \ifnum\MT@stretch=\z@
      \ifnum\MT@shrink=\z@
        \MT@warning@nl{%
          Both the stretch and shrink limit are set to zero.\MessageBreak
          Disabling font expansion}%
        \MT@expansionfalse
      \fi
    \fi
  \fi
  \ifMT@expansion
    \edef\MT@active@features{\MT@active@features,ex}%
    \pdfadjustspacing\MT@ex@level
    \MT@info@nl{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
                (level \number\MT@ex@level),\MessageBreak
                stretch: \number\MT@stretch, shrink: \number\MT@shrink,
                step: \number\MT@step, \ifMT@selected\else non-\fi selected}%
```

`\MT@check@step` Check whether `stretch` and `shrink` are multiples of `step`.

```
    \def\MT@check@step#1{%
      \@tempcnta=\csname MT@#1\endcsname
      \divide\@tempcnta \MT@step
      \multiply\@tempcnta \MT@step
      \ifnum\@tempcnta=\csname MT@#1\endcsname\else
        \MT@warning@nl{The #1 amount is not a multiple of step.\MessageBreak
                       The effective maximum #1 is \the\@tempcnta\space
                       (step \number\MT@step)}%
      \fi
    }%
    \MT@check@step{stretch}%
    \MT@check@step{shrink}%
    \MT@check@active@set{ex}%
```

Inside `\showhyphens`, font expansion should be disabled.

```
    \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
      \color@begingroup\everypar{}\parfillskip\z@skip
      \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
      \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}%
```

`\showhyphens` I wonder why it's defined globally (in `ltfssbas.dtx`)?

```
    \gdef\showhyphens#1{\setbox0\vbox{%
      \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
      \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
      \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}%

  \else
    \let\MT@expansion\relax
    \MT@info@nl{No font expansion}%
  \fi
}
\MT@requires@pdftex6{
```

`\MT@warn@lua` Switch off the features that don't work with luaTEX.

```
⟨*lua⟩
  \def\MT@warn@lua#1{%
    \MT@error{The `#1' feature doesn't currently work\MessageBreak with luatex}
             {Use pdftex instead.}%
```

```
    \csname MT@#1false\endcsname
    \MT@let@nc{MT@#1}\relax
  }
⟨/lua⟩
⟨/package⟩
  \MT@addto@setup{%
⟨*package⟩
```

Tracking, spacing and kerning.

```
    \ifMT@tracking
⟨lua⟩       \MT@requires@luatex{\MT@warn@lua{tracking}}{%
        \edef\MT@active@features{\MT@active@features,tr}%
        \MT@info@nl{Tracking enabled}%
        \MT@check@active@set{tr}%
```

Enable protrusion for compensation at the line edges.

```
        \ifMT@protrusion\else\pdfprotrudechars\@ne\fi
⟨lua⟩       }%
    \else
      \let\MT@tracking\relax
      \MT@info@nl{No tracking}%
    \fi
    \ifMT@spacing
⟨lua⟩       \MT@requires@luatex{\MT@warn@lua{spacing}}{%
        \edef\MT@active@features{\MT@active@features,sp}%
        \pdfadjustinterwordglue\@ne
        \MT@info@nl{Adjustment of interword spacing enabled}%
        \MT@check@active@set{sp}%
⟨lua⟩       }%
    \else
      \let\MT@spacing\relax
      \MT@info@nl{No adjustment of interword spacing}%
    \fi
    \ifMT@kerning
⟨lua⟩       \MT@requires@luatex{\MT@warn@lua{kerning}}{%
        \edef\MT@active@features{\MT@active@features,kn}%
        \pdfprependkern\@ne
        \pdfappendkern\@ne
        \MT@info@nl{Adjustment of character kerning enabled}%
        \MT@check@active@set{kn}%
⟨lua⟩       }%
    \else
      \let\MT@kerning\relax
      \MT@info@nl{No adjustment of character kerning}%
    \fi
⟨/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.

```
    \ifnum\pdfoutput<\@ne
      \def\MT@warn@tracking@DVI{%
        \MT@warning@nl{%
            You are using tracking/letterspacing in DVI mode.\MessageBreak
            This will probably not work, unless the post-\MessageBreak
            processing program (dvips, dvipdfm(x), ...) is\MessageBreak
            able to create the virtual fonts on the fly}%
        \MT@glet\MT@warn@tracking@DVI\relax
      }%
```

```
    \else
      \def\MT@warn@tracking@DVI{%
        \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
        \MT@glet\MT@warn@tracking@DVI\relax
      }%
    \fi
    \ifnum\MT@letterspace=\m@ne
      \let\MT@letterspace\MT@letterspace@default
    \else
      \MT@ls@too@large\MT@letterspace
    \fi
  }%
```

If pdfTEX is too old, we disable tracking, spacing and kerning, and throw an error message.

```
⟨*package⟩
}{
  \MT@addto@setup{%
    \ifMT@tracking
      \MT@error{Tracking only works with pdftex version 1.40\MessageBreak
        or newer. Switching it off}{Upgrade pdftex.}%
    \else
      \MT@info@nl{No tracking (pdftex too old)}%
    \fi
    \ifMT@spacing
      \MT@error{Adjustment of interword spacing only works with\MessageBreak
        pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%
    \else
      \MT@info@nl{No adjustment of interword spacing (pdftex too old)}%
    \fi
    \ifMT@kerning
      \MT@error{Character kerning only works with\MessageBreak
        pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%
    \else
      \MT@info@nl{No adjustment of character kerning (pdftex too old)}%
    \fi
  }
}
```

`\DisableLigatures` is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```
\MT@requires@pdftex5{
  \MT@addto@setup{%
    \ifMT@noligatures \else
      \let\MT@noligatures\relax
    \fi
  }
}\relax
```

Remove the leading comma in `\MT@active@features`, and set the document switch to true.

```
\MT@addto@setup{%
  \ifx\MT@active@features\@empty \else
    \edef\MT@active@features{\expandafter\@gobble\MT@active@features}%
  \fi
  \MT@documenttrue
}
```

`\MT@set@babel@context` Interaction with babel.

```
\def\MT@set@babel@context#1{%
  \MT@ifdefined@n@TF{MT@babel@#1}{%
    \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
    \expandafter\MT@exp@one@n\expandafter\microtypecontext
      \csname MT@babel@#1\endcsname
  }{%
    \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
  }%
}
```

`\MT@shorthandoff` Active characters can only be switched off if babel isn't loaded after microtype.

```
\@ifpackageloaded{babel}{
  \def\MT@shorthandoff#1#2{%
    \MT@info@nl{Switching off #1 babel's active characters (#2)}%
    \shorthandoff{#2}}
}{
  \def\MT@shorthandoff#1#2{%
    \MT@error{You must load `babel' before `\MT@MT'}
             {Otherwise, `\MT@MT' cannot switch off #1 babel's\MessageBreak
              active characters.}}
}
```

We patch the language switching commands to enable language-dependent setup.

```
\MT@addto@setup{%
  \ifMT@babel
    \@ifpackageloaded{babel}{%
      \MT@info@nl{Redefining babel's language switching commands}%
      \let\MT@orig@select@language\select@language
      \def\select@language#1{%
        \MT@orig@select@language{#1}%
        \MT@set@babel@context{#1}%
      }%
      \let\MT@orig@foreign@language\foreign@language
      \def\foreign@language#1{%
        \MT@orig@foreign@language{#1}%
        \MT@set@babel@context{#1}%
      }%
      \ifMT@kerning
```

Disable French babel's active characters.

```
        \MT@if@false
        \MT@with@babel@and@T{french}  \MT@if@true
        \MT@with@babel@and@T{frenchb} \MT@if@true
        \MT@with@babel@and@T{francais}\MT@if@true
        \MT@with@babel@and@T{canadien}\MT@if@true
        \MT@with@babel@and@T{acadian} \MT@if@true
        \ifMT@if@\MT@shorthandoff{French}{:;!?}\fi
```

Disable Turkish babel's active characters.

```
        \MT@if@false
        \MT@with@babel@and@T{turkish} \MT@if@true
        \ifMT@if@\MT@shorthandoff{Turkish}{:!=}\fi
      \fi
```

In case babel was loaded before microtype:

```
      \MT@set@babel@context\languagename
    }{%
      \MT@warning@nl{You did not load the babel package.\MessageBreak
        The `babel' option won't have any effect}%
    }%
```

```
  \fi
}
```

Now we close the `\fi` from `\ifMT@draft`.

```
\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.

```
  \selectfont}
```

`\MT@curr@file` This is the current file (hopefully with the correct extension).

```
\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).

```
⟨/package⟩
⟨plain⟩\MT@requires@latex1{
\AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@\@empty}
⟨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).

```
⟨*package⟩
\MT@requires@pdftex6{
  \AtBeginDocument{%
    \ifMT@spacing
      \ifMT@babel \else
        \ifnum\sfcode`\. > 1500
          \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
            \MT@warning@nl{%
              \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
              interword spacing will disable it. You might want\MessageBreak
              to add `\@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
              to your preamble}%
          }%
        \fi
      \fi
    \fi
  }
}\relax
⟨/package⟩
```

Restore catcodes.

```
\MT@restore@catcodes
```

That was that.

```
⟨/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: `ddtbaj$rob$1@online.de`

⟨∗config⟩

15.1 Font sets

We first declare some sets in the main configuration file.

```
⟨∗m-t⟩
%%% ------------------------------------------------------------------------
%%% FONT SETS

\DeclareMicrotypeSet{all}
   { }


\DeclareMicrotypeSet{allmath}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,OML,OMS,U} }

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


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

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


\DeclareMicrotypeSet{smallcaps}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1},
     shape    = {sc*}
   }


\DeclareMicrotypeSet{footnotesize}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1},
     size     = {-small}
   }

\DeclareMicrotypeSet{scriptsize}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1},
     size     = {-footnotesize}
   }

\DeclareMicrotypeSet{normalfont}
   { font = */*/*/*/* }
```

The default sets.

```
%%% ------------------------------------------------------------------------
```

```
%%% DEFAULT SETS

\DeclareMicrotypeSetDefault[protrusion]{alltext}
\DeclareMicrotypeSetDefault[expansion] {basictext}
\DeclareMicrotypeSetDefault[spacing]   {basictext}
\DeclareMicrotypeSetDefault[kerning]   {alltext}
\DeclareMicrotypeSetDefault[tracking]  {smallcaps}

```

15.2 Font variants and aliases

```
%%% ------------------------------------------------------------------------
%%% FONT VARIANTS AND ALIASES

```

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals):

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

```

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.

```
\DeclareMicrotypeAlias{lmr} {cmr}  % lmodern
\DeclareMicrotypeAlias{aer} {cmr}  % ae
\DeclareMicrotypeAlias{zer} {cmr}  % zefonts
\DeclareMicrotypeAlias{cmor}{cmr}  % eco
\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).

```
\DeclareMicrotypeAlias{pxr} {ppl}  % pxfonts
\DeclareMicrotypeAlias{qpl} {ppl}  % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)
```

The 'FPL Neu' fonts, a 're-implementation' of Palatino.

```
\DeclareMicrotypeAlias{fp9x}{pplx} % FPL Neu
\DeclareMicrotypeAlias{fp9j}{pplj} %   "
\DeclareMicrotypeAlias{txr} {ptm}  % txfonts
\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.

```
\DeclareMicrotypeAlias{zeur}{eur}  % Euler VM
\DeclareMicrotypeAlias{zeus}{eus}  %   "
```

MicroPress's Charter version (chmath).

```
\DeclareMicrotypeAlias{chr} {bch}  % CH Math
```

The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

```
\DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
\DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
\DeclareMicrotypeAlias{ulg} {blg}  % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

Euro symbol fonts, to save some files.

```
\DeclareMicrotypeAlias{zpeus}   {zpeu}    % Adobe Euro sans -> serif
\DeclareMicrotypeAlias{eurosans}{zpeu}    % Adobe Euro sans -> serif
\DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif

```

15.3 Interaction with babel

Contexts that are to be set when switching to a language.

```
%%% ------------------------------------------------------------------------
%%% INTERACTION WITH THE 'babel' PACKAGE

\DeclareMicrotypeBabelHook
   {english,UKenglish,british,USenglish,american}
   {kerning=, spacing=nonfrench}

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

\DeclareMicrotypeBabelHook
   {turkish}
   {kerning=turkish, spacing=}

```

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 `\mathchardef`ed 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.

```
⟨/m-t⟩
⟨∗m-t | zpeu | mvs⟩
%%% ------------------------------------------------------------------------
%%% CHARACTER INHERITANCE

⟨/m-t | zpeu | mvs⟩
⟨∗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.

```
\DeclareCharacterInheritance
   { encoding = OT1 }
   { f = {011}, % ff
     i = {\i},
     j = {\j},
     O = {\O},
     o = {\o}
   }

```

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.

```
\DeclareCharacterInheritance
   { encoding = T1 }
   { A = {\‘A,\’A,\^A,\~A,\"A,\r A,\k A,\u A},
     a = {\‘a,\’a,\^a,\~a,\"a,\r a,\k a,\u a},
     C = {\’C,\c C,\v C},
     c = {\’c,\c c,\v c},
     D = {\v D,\DH},
     d = {\v d,\dj},
     E = {\‘E,\’E,\^E,\"E,\k E,\v E},
     e = {\‘e,\’e,\^e,\"e,\k e,\v e},
     f = {027}, % ff
     G = {\u G},
     g = {\u g},
     I = {\‘I,\’I,\^I,\"I,\.I},
     i = {\‘i,\’i,\^i,\"i,\i},
     j = {\j},
     L = {\L,\’L,\v L},
     l = {\l,\’l,\v l},
     N = {\’N,\~N,\v N},
     n = {\’n,\~n,\v n},
     O = {\O,\‘O,\’O,\^O,\~O,\"O,\H O},
     o = {\o,\‘o,\’o,\^o,\~o,\"o,\H o},
     R = {\’R,\v R},
     r = {\’r,\v r},
     S = {\’S,\c S,\v S,\SS},
     s = {\’s,\c s,\v s},
     T = {\c T,\v T},
```

```
    t = {\c t,\v t},
    U = {\‘U,\’U,\^U,\"U,\H U,\r U},
    u = {\‘u,\’u,\^u,\"u,\H u,\r u},
    Y = {\’Y,\"Y},
    y = {\’y,\"y},
    Z = {\’Z,\.Z,\v Z},
    z = {\’z,\.z,\v z}
```

The ‘soft hyphen’ often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```
%   - = {127},
  }
```

15.5.3 LY1

More characters: `008` (‘fl’), `012` (‘fi’), `014` (‘ffi’), `015` (‘ffl’), Æ, æ, Œ, œ.

```
\DeclareCharacterInheritance
  { encoding = LY1 }
  { A = {\‘A,\’A,\^A,\~A,\"A,\r A},
    a = {\‘a,\’a,\^a,\~a,\"a,\r a},
    C = {\c C},
    c = {\c c},
    D = {\DH},
    E = {\‘E,\’E,\^E,\"E},
    e = {\‘e,\’e,\^e,\"e},
    f = {011}, % ff
    I = {\‘I,\’I,\^I,\"I},
    i = {\‘i,\’i,\^i,\"i,\i},
    L = {\L},
    l = {\l},
    N = {\~N},
    n = {\~n},
    O = {\‘O,\’O,\^O,\~O,\"O,\O},
    o = {\‘o,\’o,\^o,\~o,\"o,\o},
    S = {\v S},
    s = {\v s},
    U = {\‘U,\’U,\^U,\"U},
    u = {\‘u,\’u,\^u,\"u},
    Y = {\’Y,\"Y},
    y = {\’y,\"y},
    Z = {\v Z},
    z = {\v z}
  }
```

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.

```
\DeclareCharacterInheritance
  { encoding = OT4 }
  { A = {\k A},
    a = {\k a},
    C = {\’C},
    c = {\’c},
    E = {\k E},
    e = {\k e},
```

```
    f = {011}, % ff
    i = {\i},
    j = {\j},
    L = {\L},
    l = {\l},
    N = {\'N},
    n = {\'n},
    O = {\O,\'O},
    o = {\o,\'o},
    S = {\'S},
    s = {\'s},
    Z = {\'Z,\.Z},
    z = {\'z,\.z}
  }
```

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.

```
\DeclareCharacterInheritance
  { encoding = QX }
  { A = {\`A,\'A,\^A,\~A,\"A,\k A,\AA},
    a = {\`a,\'a,\^a,\~a,\"a,\k a,\aa},
    C = {\'C,\c C},
    c = {\'c,\c c},
    D = {\DH},
    E = {\`E,\'E,\^E,\"E,\k E},
    e = {\`e,\'e,\^e,\"e,\k e},
    f = {011}, % ff
    I = {\`I,\'I,\^I,\"I,\k I},
    i = {\`i,\'i,\^i,\"i,\k i,\i},
    j = {\j},
    L = {\L},
    l = {\l},
    N = {\'N,\~N},
    n = {\'n,\~n},
    O = {\O,\`O,\'O,\^O,\~O,\"O},
    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.

```
    S = {\'S,\c S,\textcommabelow S,\v S},
    s = {\'s,\c s,\textcommabelow s,\v s},
    T = {\c T,\textcommabelow T},
    t = {\c t,\textcommabelow t},
    U = {\`U,\'U,\^U,\"U,\k U},
    u = {\`u,\'u,\^u,\"u,\k u},
    Y = {\'Y,\"Y},
    y = {\'y,\"y},
    Z = {\'Z,\.Z,\v Z},
    z = {\'z,\.z,\v z},
    . = \textellipsis
  }
```

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.

```
\DeclareCharacterInheritance
   { encoding = T5 }
   { A = {\‘A,\’A,\~A,\h A,\d A,\^A,\u A,
          \‘\Acircumflex,\’\Acircumflex,\~\Acircumflex,\h\Acircumflex,\d\Acircumflex,
          \‘\Abreve,\’\Abreve,\~\Abreve,\h\Abreve,\d\Abreve},
     a = {\‘a,\’a,\~a,\h a,\d a,\^a,\u a,
          \‘\acircumflex,\’\acircumflex,\~\acircumflex,\h\acircumflex,\d\acircumflex,
          \‘\abreve,\’\abreve,\~\abreve,\h\abreve,\d\abreve},
     D = {\DJ},
     d = {\dj},
     E = {\‘E,\’E,\~E,\h E,\d E,\^E,
          \‘\Ecircumflex,\’\Ecircumflex,\~\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
     e = {\‘e,\’e,\~e,\h e,\d e,\^e,
          \‘\ecircumflex,\’\ecircumflex,\~\ecircumflex,\h\ecircumflex,\d\ecircumflex},
     I = {\‘I,\’I,\~I,\h I,\d I},
     i = {\‘i,\’i,\~i,\h i,\d i,\i},
     O = {\‘O,\’O,\~O,\h O,\d O,\^O,\horn O,
          \‘\Ocircumflex,\’\Ocircumflex,\~\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
          \‘\Ohorn,\’\Ohorn,\~\Ohorn,\h\Ohorn,\d\Ohorn},
     o = {\‘o,\’o,\~o,\h o,\d o,\^o,\horn o,
          \‘\ocircumflex,\’\ocircumflex,\~\ocircumflex,\h\ocircumflex,\d\ocircumflex,
          \‘\ohorn,\’\ohorn,\~\ohorn,\h\ohorn,\d\ohorn},
     U = {\‘U,\’U,\~U,\h U,\d U,\horn U,
          \‘\Uhorn,\’\Uhorn,\~\Uhorn,\h\Uhorn,\d\Uhorn},
     u = {\‘u,\’u,\~u,\h u,\d u,\horn u,
          \‘\uhorn,\’\uhorn,\~\uhorn,\h\uhorn,\d\uhorn},
     Y = {\‘Y,\’Y,\~Y,\h Y,\d Y},
     y = {\‘y,\’y,\~y,\h y,\d y}
   }

⟨/m-t⟩
```

15.5.7 Euro symbols

Make Euro symbols settings simpler.

```
⟨*zpeu⟩
\DeclareCharacterInheritance
   { encoding = U,
     family   = {zpeu,zpeus,eurosans} }
   { E = 128 }

⟨/zpeu⟩
⟨*mvs⟩
\DeclareCharacterInheritance
   { encoding = OT1,
     family   = mvs }
   { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm

```

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.

```
\DeclareCharacterInheritance
   { encoding = U,
```

```
    family   = mvs }
  { 164 = {099,100,101} }

⟨/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.

```
⟨*m-t⟩
%%% ------------------------------------------------------------------------
%%% TRACKING/LETTERSPACING

\SetTracking
  [ name         = default,
    no ligatures = {f} ]
  { encoding     = {OT1,T1,T2A,LY1,OT4,QX} }
  { }

```

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).

```
%%% ------------------------------------------------------------------------
%%% EXPANSION

\SetExpansion
   [ name     = default      ]
   { encoding = {OT1,OT4,QX,T1,LY1} }
   {
     A = 500,     a = 700,
   \AE = 500,   \ae = 700,
     B = 700,     b = 700,
     C = 700,     c = 700,
     D = 500,     d = 700,
     E = 700,     e = 700,
     F = 700,
     G = 500,     g = 700,
     H = 700,     h = 700,
     K = 700,     k = 700,
     M = 700,     m = 700,
     N = 700,     n = 700,
     O = 500,     o = 700,
   \OE = 500,   \oe = 700,
     P = 700,     p = 700,
     Q = 500,     q = 700,
     R = 700,
     S = 700,     s = 700,
     U = 700,     u = 700,
     W = 700,     w = 700,
     Z = 700,     z = 700,
     2 = 700,
     3 = 700,
     6 = 700,
     8 = 700,
     9 = 700
```

```
  }
```

Settings for Cyrillic T2A encoding.[18]

```
\SetExpansion
  [ name     = T2A ]
  { encoding = T2A }
  {
    A = 500,     a = 700,
    B = 700,     b = 700,
    C = 700,     c = 700,
    D = 500,     d = 700,
    E = 700,     e = 700,
    F = 700,
    G = 500,     g = 700,
    H = 700,     h = 700,
    K = 700,     k = 700,
    M = 700,     m = 700,
    N = 700,     n = 700,
    O = 500,     o = 700,
    P = 700,     p = 700,
    Q = 500,     q = 700,
    R = 700,
    S = 700,     s = 700,
    U = 700,     u = 700,
    W = 700,     w = 700,
    Z = 700,     z = 700,
    2 = 700,
    3 = 700,
    6 = 700,
    8 = 700,
    9 = 700,
    \CYRA = 500,     \cyra = 700,
    \CYRB = 700,     \cyrb = 700,
    \CYRV = 700,     \cyrv = 700,
    \CYRG = 700,     \cyrg = 700,
    \CYRD = 700,     \cyrd = 700,
    \CYRE = 700,     \cyre = 700,
    \CYRZH = 700,    \cyrzh = 700,
    \CYRZ = 700,     \cyrz = 700,
    \CYRI = 700,     \cyri = 700,
    \CYRISHRT = 700, \cyrishrt = 700,
    \CYRK = 700,     \cyrk = 700,
    \CYRL = 700,     \cyrl = 700,
    \CYRM = 700,     \cyrm = 700,
    \CYRN = 700,     \cyrn = 700,
    \CYRO = 500,     \cyro = 700,
    \CYRP = 700,     \cyrp = 700,
    \CYRR = 700,     \cyrr = 700,
    \CYRS = 700,     \cyrs = 700,
    \CYRT = 700,     \cyrt = 700,
    \CYRU = 700,     \cyru = 700,
    \CYRF = 700,     \cyrf = 700,
    \CYRH = 700,     \cyrh = 700,
    \CYRC = 700,     \cyrc = 700,
    \CYRCH = 700,    \cyrch = 700,
    \CYRSH = 700,    \cyrsh = 700,
    \CYRSHCH = 700,  \cyrshch = 700,
```

18 Contributed by *Karl Karlsson*.

```
    \CYRHRDSN = 700, \cyrhrdsn = 700,
    \CYRERY = 700,   \cyrery = 700,
    \CYRSFTSN = 700, \cyrsftsn = 700,
    \CYREREV = 700,  \cyrerev = 700,
    \CYRYU = 700,    \cyryu = 700,
    \CYRYA = 700,    \cyrya = 700
  }

```

T5 encoding does not contain \AE, \ae, \OE and \oe.

```
\SetExpansion
  [ name     = T5 ]
  { encoding = T5 }
  {
    A = 500,     a = 700,
    B = 700,     b = 700,
    C = 700,     c = 700,
    D = 500,     d = 700,
    E = 700,     e = 700,
    F = 700,
    G = 500,     g = 700,
    H = 700,     h = 700,
    K = 700,     k = 700,
    M = 700,     m = 700,
    N = 700,     n = 700,
    O = 500,     o = 700,
    P = 700,     p = 700,
    Q = 500,     q = 700,
    R = 700,
    S = 700,     s = 700,
    U = 700,     u = 700,
    W = 700,     w = 700,
    Z = 700,     z = 700,
    2 = 700,
    3 = 700,
    6 = 700,
    8 = 700,
    9 = 700
  }

⟨/m-t⟩
```

15.8 Character protrusion

```
%%% ------------------------------------------------------------------------
%%% PROTRUSION

```

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*.

```
⟨ugm⟩    [ name      = ugm-default ]
⟨m-t | cmr | pmn⟩    { }
⟨bch | blg | pad | ugm⟩    { encoding = OT1,
⟨ppl | ptm⟩    { encoding = {OT1,OT4},
⟨bch⟩      family    = bch }
⟨blg⟩      family    = blg }
⟨pad⟩      family    = {pad,padx,padj} }
⟨ppl⟩      family    = {ppl,pplx,pplj} }
⟨ptm⟩      family    = {ptm,ptmx,ptmj} }
⟨ugm⟩      family    = ugm }
   {
⟨m-t | bch | blg | cmr | pad | pmn | ppl | ptm⟩      A = {50,50},
⟨ugm⟩      A = {50,100},
⟨pad | ptm⟩    \AE = {50,  },
⟨ugm⟩    \AE = {150,50},
⟨ugm⟩      B = {  ,50},
⟨bch | pad | pmn | ugm⟩      C = {50,  },
⟨bch | pad | pmn⟩      D = {  ,50},
⟨ugm⟩      D = {  ,70},
⟨ugm⟩      E = {  ,50},
⟨m-t | bch | cmr | pad | pmn | ptm⟩      F = {  ,50},
⟨ugm⟩      F = {  ,70},
⟨bch | pad | pmn⟩      G = {50,  },
⟨ugm⟩      G = {50,50},
⟨blg⟩      I = {150,150},
⟨m-t | cmr | pad | pmn | ppl | ptm | ugm⟩      J = {50,  },
⟨bch | blg⟩      J = {100,  },
⟨!blg⟩      K = {  ,50},
⟨blg⟩      K = {50,  },
⟨m-t | bch | cmr | pad | pmn | ppl⟩      L = {  ,50},
⟨blg⟩      L = {  ,150},
⟨ptm⟩      L = {  ,80},
⟨ugm⟩      L = {  ,120},
⟨bch | pad | pmn | ugm⟩      O = {50,50},
⟨pad⟩    \OE = {50,  },
⟨ugm⟩    \OE = {50,50},
⟨blg⟩      P = {  ,100},
⟨ugm⟩      P = {  ,50},
⟨bch | pad | pmn⟩      Q = {50,70},
⟨ugm⟩      Q = {50,50},
⟨bch⟩      R = {  ,50},
⟨ugm⟩      R = {  ,70},
⟨m-t | bch | cmr | pad | pmn | ppl | ptm⟩      T = {50,50},
⟨blg⟩      T = {100,100},
⟨ugm⟩      T = {70,70},
⟨m-t | bch | cmr | pad | pmn | ppl | ptm⟩      V = {50,50},
⟨blg | ugm⟩      V = {70,70},
⟨m-t | bch | cmr | pad | pmn | ppl | ptm⟩      W = {50,50},
⟨ugm⟩      W = {70,70},
⟨m-t | bch | cmr | pad | pmn | ppl | ptm⟩      X = {50,50},
⟨ugm⟩      X = {50,70},
⟨m-t | bch | cmr | pad | pmn | ppl⟩      Y = {50,50},
⟨blg | ptm | ugm⟩      Y = {80,80},
⟨ugm⟩      Z = {50,50},
⟨blg⟩      f = {150,100},
⟨blg⟩      i = {150,150},
⟨blg⟩      j = {100,100},
⟨m-t | bch | cmr | pad | pmn | ppl | ptm⟩      k = {  ,50},
⟨ugm⟩      k = {  ,70},
⟨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},
```

```
⟨m-t | bch | pad | pmn | ptm⟩       ; = { ,300},
⟨blg⟩       ; = {200,300},
⟨cmr | ppl⟩       ; = { ,500},
⟨ugm⟩       ; = { ,400},
⟨!blg⟩       ! = { ,100},
⟨blg⟩       ! = {200,200},
⟨m-t | pad | pmn | ptm⟩       ? = { ,100},
⟨bch | cmr | ppl | ugm⟩       ? = { ,200},
⟨blg⟩       ? = {150,150},
⟨pmn⟩       " = {300,300},
⟨m-t | bch | cmr | pad | pmn | ppl⟩       @ = {50,50},
⟨ptm⟩       @ = {100,100},
⟨m-t | bch | blg | cmr | pad | pmn | ppl | ptm⟩       ~ = {200,250},
⟨ugm⟩       ~ = {300,350},
⟨pad | ppl | ptm⟩       & = {50,100},
⟨ugm⟩       & = {  ,100},
⟨m-t | cmr | pad | pmn⟩       \% = {50,50},
⟨bch⟩       \% = {  ,50},
⟨ppl | ptm⟩       \% = {100,100},
⟨ugm⟩       \% = {50,100},
⟨blg⟩       \# = {100,100},
⟨m-t | ppl | ptm | ugm⟩       * = {200,200},
⟨bch | pmn⟩       * = {200,300},
⟨blg⟩       * = {150,200},
⟨cmr | pad⟩       * = {300,300},
⟨m-t | cmr | ppl | ptm⟩       + = {250,250},
⟨bch⟩       + = {150,250},
⟨pad⟩       + = {300,300},
⟨blg | pmn⟩       + = {150,200},
⟨ugm⟩       + = {250,300},
⟨blg | ugm⟩       {=}= {200,200},
⟨m-t | pad | pmn | ptm⟩       ( = {100,   },     ) = {   ,200},
⟨bch | ugm⟩       ( = {200,   },     ) = {   ,200},
⟨cmr | blg⟩       ( = {300,   },     ) = {   ,300},
⟨ppl⟩       ( = {100,   },     ) = {   ,300},
⟨bch | pmn⟩       [ = {100,   },     ] = {   ,100},
⟨blg⟩       [ = {300,100},     ] = {   ,300},

⟨m-t | pad | pmn | ptm⟩       / = {100,200},
⟨bch⟩       / = { ,200},
⟨blg⟩       / = {300,300},
⟨cmr | ppl⟩       / = {200,300},
⟨ugm⟩       / = {100,300},
⟨m-t | ptm⟩       - = {500,500},
⟨bch | cmr | ppl⟩       - = {400,500},
⟨blg⟩       - = {300,400},
⟨pad⟩       - = {300,500},
⟨pmn⟩       - = {200,400},
⟨ugm⟩       - = {500,600},
⟨blg⟩       < = {200,100},     > = {100,200},
⟨blg⟩       _ = {150,250},
⟨blg⟩       | = {250,250},
⟨m-t | pmn⟩       \textendash       = {200,200},   \textemdash       = {150,150},
⟨bch⟩       \textendash       = {200,300},   \textemdash       = {150,250},
⟨cmr⟩       \textendash       = {400,300},   \textemdash       = {300,200},
⟨pad | ppl | ptm⟩       \textendash       = {300,300},   \textemdash       = {200,200},
⟨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).

```
⟨m-t | bch | pmn⟩     \textquoteleft    = {300,400},   \textquoteright    = {300,400},
⟨blg⟩      \textquoteleft     = {400,600},   \textquoteright    = {400,600},
⟨cmr⟩      \textquoteleft    = {500,700},   \textquoteright    = {500,600},
⟨pad | ppl⟩     \textquoteleft    = {500,700},   \textquoteright    = {500,700},
⟨ptm⟩      \textquoteleft    = {500,500},   \textquoteright    = {300,500},
⟨ugm⟩      \textquoteleft    = {300,600},   \textquoteright    = {300,600},
⟨m-t | bch | pmn⟩     \textquotedblleft = {300,300},   \textquotedblright = {300,300}
⟨blg⟩      \textquotedblright = {300,400}
⟨cmr⟩      \textquotedblleft = {500,300},   \textquotedblright = {200,600}
⟨pad | ppl | ptm⟩     \textquotedblleft = {300,400},   \textquotedblright = {300,400}
⟨ugm⟩      \textquotedblleft = {400,400},   \textquotedblright = {400,400}
   }

```

Greek uppercase letters are in OT1 encoding only.

```
⟨*m-t | cmr | pmn⟩
\SetProtrusion
⟨m-t⟩    [ name     = OT1-default,
⟨cmr⟩    [ name     = cmr-OT1,
⟨pmn⟩    [ name     = pmnj-OT1,
⟨m-t⟩      load     = default ]
⟨cmr⟩      load     = cmr-default ]
⟨pmn⟩      load     = pmnj-default ]
⟨m-t⟩    { encoding = OT1 }
⟨cmr⟩    { encoding = {OT1,OT4},
⟨pmn⟩    { encoding = OT1,
⟨cmr⟩      family   = cmr  }
⟨pmn⟩      family   = pmnj }
   {
⟨m-t | cmr⟩     \AE = {50,   },
⟨pmn⟩      \OE = {50,  }
⟨*cmr⟩
     "00 = {   ,150}, % \Gamma
     "01 = {100,100}, % \Delta
     "02 = { 50, 50}, % \Theta
     "03 = {100,100}, % \Lambda
     "06 = { 50, 50}, % \Sigma
     "07 = {100,100}, % \Upsilon
     "08 = { 50, 50}, % \Phi
     "09 = { 50, 50}  % \Psi
```

Remaining slots can be found in the source file.

```
⟨/cmr⟩
   }

⟨/m-t | cmr | pmn⟩
```

T1 and LY1 encodings contain some more characters. The default list will be loaded first.

```
\SetProtrusion
⟨m-t⟩    [ name     = T1-default,
⟨bch⟩    [ name     = bch-T1,
⟨blg⟩    [ name     = blg-T1,
⟨cmr⟩    [ name     = cmr-T1,
⟨pad⟩    [ name     = pad-T1,
⟨pmn⟩    [ name     = pmnj-T1,
⟨ppl⟩    [ name     = ppl-T1,
⟨ptm⟩    [ name     = ptm-T1,
⟨ugm⟩    [ name     = ugm-T1,
⟨m-t⟩      load     = default     ]
```

```
⟨bch⟩      load     = bch-default ]
⟨blg⟩      load     = blg-default ]
⟨cmr⟩      load     = cmr-default ]
⟨pad⟩      load     = pad-default ]
⟨pmn⟩      load     = pmnj-default ]
⟨ppl⟩      load     = ppl-default ]
⟨ptm⟩      load     = ptm-default ]
⟨ugm⟩      load     = ugm-default ]
⟨m-t⟩    { encoding = {T1,LY1}    }
⟨bch | cmr | pad | pmn | ppl⟩    { encoding = {T1,LY1},
⟨blg | ptm | ugm⟩    { encoding = {T1},
⟨bch⟩      family   = bch }
⟨blg⟩      family   = blg }
⟨cmr⟩      family   = cmr }
⟨pad⟩      family   = {pad,padx,padj} }
⟨pmn⟩      family   = pmnj }
⟨ppl⟩      family   = {ppl,pplx,pplj} }
⟨ptm⟩      family   = {ptm,ptmx,ptmj} }
⟨ugm⟩      family   = ugm }
    {
⟨m-t | cmr⟩      \AE = {50,  },
⟨bch | pmn⟩      \OE = {50,  },
⟨pmn⟩      \TH = {  ,50},
⟨blg⟩      \v L = {   ,250},
⟨blg⟩      \v d = {   ,250},
⟨blg⟩      \v l = {   ,250},
⟨blg⟩      \v t = {   ,250},
⟨blg⟩      127 =  {300,400},
⟨blg⟩      156 =  {100,   }, % IJ
⟨blg⟩      188 =  { 80, 80}, % ij
⟨m-t | bch | pad | pmn | ppl | ptm⟩        _ = {100,100},
⟨cmr⟩        _ = {200,200},
⟨ugm⟩        _ = {100,200},
⟨m-t | pad | pmn | ptm⟩      \textbackslash    = {100,200},
⟨bch⟩      \textbackslash    = {150,200},
⟨blg⟩      \textbackslash    = {250,300},
⟨cmr | ppl⟩      \textbackslash    = {200,300},
⟨ugm⟩      \textbackslash    = {100,300},
⟨ugm⟩      \textbar          = {200,200},
⟨blg⟩      \textendash       = {300,300},   \textemdash        = {150,150},
⟨blg⟩      \textquotedbl     = {300,400},   \textquotedblleft  = {300,400},
⟨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.

```
⟨m-t | cmr | pad | ppl | ptm | ugm⟩      \quotesinglbase   = {400,400},   \quotedblbase      = {400,400},
⟨blg⟩      \quotesinglbase   = {400,400},   \quotedblbase      = {300,400},
⟨bch | pmn⟩      \quotesinglbase   = {400,400},   \quotedblbase      = {300,300},
⟨m-t | bch | pmn⟩      \guilsinglleft    = {400,300},   \guilsinglright    = {300,400},
⟨blg⟩      \guilsinglleft    = {300,500},   \guilsinglright    = {300,500},
⟨cmr | pad | ppl | ptm⟩      \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
⟨ugm⟩      \guilsinglleft    = {400,400},   \guilsinglright    = {300,600},
⟨m-t⟩      \guillemotleft    = {200,200},   \guillemotright    = {200,200},
⟨cmr⟩      \guillemotleft    = {300,200},   \guillemotright    = {100,400},
⟨bch | pmn⟩      \guillemotleft    = {200,200},   \guillemotright    = {150,300},
⟨blg | pad | ppl | ptm⟩      \guillemotleft    = {300,300},   \guillemotright    = {200,400},
⟨ugm⟩      \guillemotleft    = {300,400},   \guillemotright    = {300,400},
⟨m-t | bch | cmr | pad | pmn | ppl | ugm⟩      \textexclamdown   = {100,   },   \textquestiondown  = {100,   },
⟨blg⟩      \textexclamdown   = {200,   },   \textquestiondown  = {100,   },
⟨ptm⟩      \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
```

```
⟨m-t | cmr | pad | ppl | ptm | ugm⟩      \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨bch | blg | pmn⟩      \textbraceleft    = {200,   },   \textbraceright    = {   ,300},
⟨m-t | bch | cmr | pad | ppl | ptm | ugm⟩      \textless         = {200,100},   \textgreater       = {100,200}
⟨pmn⟩      \textless         = {100,   },   \textgreater       = {   ,100},
⟨pmn⟩      \textvisiblespace = {100,100} % not in LY1
   }

```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```
⟨*cmr⟩
\SetProtrusion
   [ name     = lmr-T1,
     load     = cmr-T1   ]
   { encoding = {T1,LY1},
     family   = lmr      }
   {
     \textquotedblleft  = {300,400},  \textquotedblright = {300,400}
   }

⟨/cmr⟩
```

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion).[20]

```
⟨*m-t | cmr | pmn⟩
\SetProtrusion
⟨m-t⟩    [ name     = T2A-default,
⟨cmr⟩    [ name     = cmr-T2A,
⟨pmn⟩    [ name     = pmnj-T2A,
⟨m-t⟩      load     = default     ]
⟨cmr⟩      load     = cmr-default ]
⟨pmn⟩      load     = pmnj-default ]
   { encoding = T2A,
⟨m-t⟩    }
⟨cmr⟩      family   = cmr }
⟨pmn⟩      family   = pmnj }
   {
     \CYRA = {50,50},
     \CYRG = {  ,50},
     \CYRK = {  ,50},
     \CYRT = {50,50},
     \CYRH = {50,50},
     \CYRU = {50,50},
⟨pmn⟩      \CYRS = {50,  },
⟨pmn⟩      \CYRO = {50,50},
     \cyrk = {  ,50},
     \cyrg = {  ,50},
     \cyrh = {50,50},
⟨m-t | pmn⟩      \cyru = {50,50},
⟨cmr⟩      \cyru = {50,70},
⟨m-t⟩        _  = {100,100},
⟨cmr⟩        _  = {200,200},
⟨m-t⟩      \textbackslash    = {100,200},   \quotedblbase      = {400,400},
⟨cmr⟩      \textbackslash    = {200,300},   \quotedblbase      = {400,400},
⟨pmn⟩      \textbackslash    = {100,200},   \quotedblbase      = {300,300},
⟨cmr⟩      \textquotedbl     = {300,300},   \textquotedblleft  = {200,600},
⟨m-t⟩      \guillemotleft    = {200,200},   \guillemotright    = {200,200},
```

20 Contributed by *Karl Karlsson*.

```
⟨cmr⟩      \guillemotleft    = {300,200},   \guillemotright    = {100,400},
⟨pmn⟩      \guillemotleft    = {200,200},   \guillemotright    = {150,300},
⟨m-t | cmr⟩      \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨pmn⟩      \textbraceleft    = {200,   },   \textbraceright    = {   ,300},
⟨m-t | cmr⟩      \textless         = {200,100},   \textgreater       = {100,200}
⟨pmn⟩      \textless         = {100,   },   \textgreater       = {   ,100}
   }

⟨/m-t | cmr | pmn⟩
```

Settings for the QX encoding (generic and Times).[21] It also includes some glyphs otherwise in TS1.

```
⟨*m-t | ptm⟩
\SetProtrusion
⟨m-t⟩   [ name     = QX-default,
⟨ptm⟩   [ name     = ptm-QX,
⟨m-t⟩     load     = default ]
⟨ptm⟩     load     = ptm-default ]
⟨m-t⟩   { encoding = QX }
⟨ptm⟩   { encoding = QX,
⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
   {
     \AE = {50,  },
⟨ptm⟩      *  = {200,200},
     {=} = {100,100},
     \textunderscore   = {100,100},
     \textbackslash    = {100,200},
     \quotedblbase     = {400,400},
⟨m-t⟩     \guillemotleft    = {200,200},   \guillemotright    = {200,200},
⟨ptm⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
     \textexclamdown   = {100,   },   \textquestiondown  = {100,   },
⟨m-t⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨ptm⟩     \textbraceleft    = {200,200},   \textbraceright    = {200,300},
     \textless         = {200,100},   \textgreater       = {100,200},
     \textminus        = {200,200},   \textdegree        = {300,300},
⟨m-t⟩     \copyright        = {100,100},   \textregistered    = {100,100}
⟨ptm⟩     \copyright        = {100,150},   \textregistered    = {100,150},
⟨ptm⟩     \textxgeq         = {   ,100},   \textxleq          = {100,   },
⟨ptm⟩     \textalpha        = {   , 50},   \textDelta         = { 70, 70},
⟨ptm⟩     \textpi           = { 50, 80},   \textSigma         = {   , 70},
⟨ptm⟩     \textmu           = {   , 80},   \texteuro          = { 50, 50},
⟨ptm⟩     \textellipsis     = {150,200},   \textasciitilde    = { 80, 80},
⟨ptm⟩     \textapprox       = { 50, 50},   \textinfty         = {100,100},
⟨ptm⟩     \textdagger       = {150,150},   \textdaggerdbl     = {100,100},
⟨ptm⟩     \textdiv          = { 50,150},   \textsection       = { 80, 80},
⟨ptm⟩     \texttimes        = {100,150},   \textpm            = { 50, 80},
⟨ptm⟩     \textbullet       = {150,150},   \textperiodcentered = {300,300},
⟨ptm⟩     \textquotesingle  = {500,500},   \textquotedbl      = {300,300},
⟨ptm⟩     \textperthousand  = {    ,50}
   }

⟨/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.

```
⟨*cmr | bch⟩
\SetProtrusion
⟨cmr⟩   [ name     = cmr-T5,
```

21 Contributed by *Maciej Eder*.

```
⟨cmr⟩     load     = cmr-default ]
⟨bch⟩   [ name     = bch-T5,
⟨bch⟩     load     = bch-default ]
   { encoding = T5,
⟨cmr⟩     family   = cmr }
⟨bch⟩     family   = bch }
   {
⟨bch⟩     _ = {100,100},
⟨bch⟩     \textbackslash    = {150,200},
⟨cmr⟩     \textbackslash    = {200,300},
⟨cmr⟩     \textquotedblleft = {200,600},
⟨cmr⟩     \textquotedbl     = {300,300},
⟨bch⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {300,300},
⟨cmr⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {400,400},
⟨bch⟩     \guilsinglleft    = {400,300},   \guilsinglright    = {300,400},
⟨cmr⟩     \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
⟨bch⟩     \guillemotleft    = {200,200},   \guillemotright    = {150,300},
⟨cmr⟩     \guillemotleft    = {300,200},   \guillemotright    = {100,400},
⟨bch⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,300},
⟨cmr⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
     \textless         = {200,100},   \textgreater       = {100,200}
   }

⟨/cmr | bch⟩
```

Minion with lining numbers.

```
⟨*pmn⟩
\SetProtrusion
   [ name     = pmnx-OT1,
     load     = pmnj-default ]
   { encoding = OT1,
     family   = pmnx }
   {
     1 = {230,180}
   }

\SetProtrusion
   [ name     = pmnx-T1,
     load     = pmnj-T1  ]
   { encoding = {T1,LY1},
     family   = pmnx     }
   {
     1 = {230,180}
   }

\SetProtrusion
   [ name     = pmnx-T2A,
     load     = pmnj-T2A  ]
   { encoding = {T2A},
     family   = pmnx     }
   {
     1 = {230,180}
   }

⟨/pmn⟩
```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```
⟨*ptm⟩
 \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  }
```

```
⟨cmr | pmn⟩    { }
   {
⟨cmr | ptm⟩      A = {100,50},
⟨pad | pmn⟩      A = {50,  },
⟨ugm⟩      A = {  ,150},
⟨ppl⟩      A = {50,50},
⟨ptm⟩    \AE = {100,  },
⟨pad | ppl⟩    \AE = {50,  },
⟨cmr | pad | ppl | ptm⟩      B = {50,  },
⟨pmn⟩      B = {20,-50},
⟨bch | ppl | ptm | ugm⟩      C = {50,  },
⟨cmr | pad⟩      C = {100, },
⟨pmn⟩      C = {50,-50},
⟨cmr | pad | ppl | ptm⟩      D = {50,50},
⟨pmn⟩      D = {20,  },
⟨cmr | pad | ppl | ptm⟩      E = {50,  },
⟨pmn⟩      E = {20,-50},
⟨cmr | pad | ptm⟩      F = {100, },
⟨pmn⟩      F = {10,  },
⟨ppl⟩      F = {50,  },
⟨bch | ppl | ptm | ugm⟩      G = {50,  },
⟨cmr | pad⟩      G = {100, },
⟨pmn⟩      G = {50,-50},
⟨cmr | pad | ppl | ptm⟩      H = {50,  },
⟨cmr | pad | ptm⟩      I = {50,  },
⟨pmn⟩      I = {20,-50},
⟨cmr | ptm⟩      J = {100, },
⟨pad⟩      J = {50,  },
⟨pmn⟩      J = {20,  },
⟨cmr | pad | ppl | ptm⟩      K = {50,  },
⟨pmn⟩      K = {20,  },
⟨cmr | pad | ppl | ptm⟩      L = {50,  },
⟨pmn⟩      L = {20,50},
⟨ugm⟩      L = {  ,100},
⟨cmr | ptm⟩      M = {50,  },
⟨pmn⟩      M = {  ,-30},
⟨cmr | ptm⟩      N = {50,  },
⟨pmn⟩      N = {  ,-30},
⟨bch | pmn | ppl | ptm⟩      O = {50,  },
⟨cmr | pad⟩      O = {100, },
⟨ugm⟩      O = {70,50},
⟨ppl | ptm⟩    \OE = {50,  },
⟨pad⟩    \OE = {100, },
⟨cmr | pad | ppl | ptm⟩      P = {50,  },
⟨pmn⟩      P = {20,-50},
⟨bch | pmn | ppl | ptm⟩      Q = {50,  },
⟨cmr | pad⟩      Q = {100, },
⟨ugm⟩      Q = {70,50},
⟨cmr | pad | ppl | ptm⟩      R = {50,  },
⟨pmn⟩      R = {20,  },
⟨bch | cmr | pad | ppl | ptm⟩      S = {50,  },
⟨pmn⟩      S = {20,-30},
⟨bch | cmr | pad | ppl | ptm⟩      $ = {50,  },
⟨pmn⟩      $ = {20,-30},
⟨bch | pmn | ugm⟩      T = {70,  },
⟨cmr | pad | ppl | ptm⟩      T = {100, },
⟨cmr | pad | ppl | ptm⟩      U = {50,  },
⟨pmn⟩      U = {50,-50},
⟨cmr | pad | pmn | ugm⟩      V = {100, },
⟨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,
```

```
⟨pmn⟩      family   = pmnj,
⟨pad⟩      family   = {pad,padx,padj},
⟨ppl⟩      family   = {ppl,pplx,pplj},
⟨ptm⟩      family   = {ptm,ptmx,ptmj},
⟨ugm⟩      family   = ugm,
⟨m-t | bch | pad | pmn | ppl | ptm⟩      shape    = {it,sl}  }
⟨blg | cmr | ugm⟩      shape    = it       }
   {
⟨m-t | bch | pmn⟩      _ = {  ,100},
⟨blg⟩      _ = {0,300},
⟨cmr | ugm⟩      _ = {100,200},
⟨pad | ppl | ptm⟩      _ = {100,100},
⟨blg⟩      . = {400,600},
⟨blg⟩     {,}= {300,500},
⟨cmr⟩      \AE = {100,   },
⟨pmn⟩      \AE = {  ,-50},
⟨bch | pmn⟩      \OE = { 50,   },
⟨cmr⟩      \OE = {100,   },
⟨pmn⟩      031 = { ,-100}, % ffl
⟨cmr | ptm⟩      156 = {100, },  % IJ
⟨pad⟩      156 = {50,  },  % IJ
⟨pmn⟩      156 = {20,  },  % IJ
⟨pmn⟩      188 = {  ,-30}, % ij
⟨pmn⟩    \v t = { ,100},
⟨m-t | pad | ppl | ptm⟩      \textbackslash    = {100,200},
⟨cmr | ugm⟩      \textbackslash    = {300,300},
⟨bch⟩      \textbackslash    = {150,150},
⟨pmn⟩      \textbackslash    = {100,150},
⟨ugm⟩      \textbar          = {200,200},
⟨cmr⟩      \textquotedblleft = {500,300},
⟨blg⟩     \textquoteleft     = {400,400},    \textquoteright    = {400,400},
⟨blg⟩     \textquotedbl      = {300,300},    \textquotedblleft  = {300,300},
⟨blg⟩     \textquotedblright = {300,300},    \quotedblbase      = {200,600},
⟨m-t | ptm⟩      \quotesinglbase   = {300,700},   \quotedblbase      = {400,500},
⟨cmr⟩      \quotesinglbase   = {300,700},   \quotedblbase      = {200,600},
⟨bch | pmn⟩      \quotesinglbase   = {200,500},   \quotedblbase      = {150,500},
⟨pad | ppl⟩      \quotesinglbase   = {500,500},   \quotedblbase      = {400,400},
⟨ugm⟩      \quotesinglbase   = {300,700},   \quotedblbase      = {300,500},
⟨m-t | ppl | ptm⟩      \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
⟨bch | pmn⟩      \guilsinglleft    = {300,400},   \guilsinglright    = {200,500},
⟨cmr⟩      \guilsinglleft    = {500,300},   \guilsinglright    = {400,400},
⟨pad⟩      \guilsinglleft    = {500,400},   \guilsinglright    = {300,500},
⟨ugm⟩      \guilsinglleft    = {400,400},   \guilsinglright    = {300,600},
⟨m-t | ppl⟩      \guillemotleft    = {300,300},   \guillemotright    = {300,300},
⟨bch | pmn⟩      \guillemotleft    = {200,300},   \guillemotright    = {150,400},
⟨cmr⟩      \guillemotleft    = {400,100},   \guillemotright    = {200,300},
⟨pad⟩      \guillemotleft    = {300,300},   \guillemotright    = {200,400},
⟨ptm⟩      \guillemotleft    = {300,400},   \guillemotright    = {200,400},
⟨ugm⟩      \guillemotleft    = {300,400},   \guillemotright    = {300,400},
⟨m-t | pad | ppl | ugm⟩      \textexclamdown   = {100,   },   \textquestiondown  = {200,   },
⟨cmr | ptm⟩      \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
⟨pmn⟩      \textexclamdown   = {-50,   },   \textquestiondown  = {-50,   },
⟨m-t | ppl | ugm⟩      \textbraceleft    = {200,100},   \textbraceright    = {200,200},
⟨bch | pmn⟩      \textbraceleft    = {200,   },   \textbraceright    = {   ,200},
⟨cmr | pad | ptm⟩      \textbraceleft    = {400,100},   \textbraceright    = {200,200},
⟨bch | pmn⟩      \textless         = {100,   },   \textgreater       = {   ,100},
⟨cmr | pad | ppl | ptm⟩      \textless         = {300,100},   \textgreater       = {200,100}
⟨pmn⟩      \textvisiblespace = {100,100}
  }
```

```
⟨∗m-t | cmr | pmn⟩
\SetProtrusion
⟨m-t⟩    [ name     = T2A-it-default,
⟨cmr⟩    [ name     = cmr-it-T2A,
⟨pmn⟩    [ name     = pmnj-it-T2A,
⟨m-t⟩      load     = OT1-it   ]
⟨cmr⟩      load     = cmr-it   ]
⟨pmn⟩      load     = pmnj-it  ]
   { encoding = T2A,
⟨cmr⟩      family   = cmr,
⟨pmn⟩      family   = pmnj,
⟨m-t | pmn⟩      shape    = {it,sl}  }
⟨cmr⟩      shape    = it       }
   {
⟨cmr⟩      \CYRA = {100,50},
⟨pmn⟩      \CYRA = {50,  },
⟨cmr⟩      \CYRB = {50,  },
⟨cmr⟩      \CYRV = {50,  },
⟨pmn⟩      \CYRV = {20,-50},
⟨cmr⟩      \CYRG = {100, },
⟨pmn⟩      \CYRG = {10,  },
⟨cmr⟩      \CYRD = {50,  },
⟨cmr⟩      \CYRE = {50,  },
⟨pmn⟩      \CYRE = {20,-50},
⟨cmr⟩      \CYRZH = {50,  },
⟨cmr⟩      \CYRZ = {50,  },
⟨pmn⟩      \CYRZ = {20,-50},
⟨cmr⟩      \CYRI = {50,  },
⟨pmn⟩      \CYRI = {  ,-30},
⟨cmr⟩      \CYRISHRT = {50,  },
⟨cmr⟩      \CYRK = {50,  },
⟨pmn⟩      \CYRK = {20,  },
⟨cmr⟩      \CYRL = {50,  },
⟨cmr⟩      \CYRM = {50,  },
⟨pmn⟩      \CYRM = {  ,-30},
⟨cmr⟩      \CYRN = {50,  },
⟨cmr⟩      \CYRO = {100, },
⟨pmn⟩      \CYRO = {50,  },
⟨cmr⟩      \CYRP = {50,  },
⟨cmr⟩      \CYRR = {50,  },
⟨pmn⟩      \CYRR = {20,-50},
⟨cmr⟩      \CYRS = {100, },
⟨pmn⟩      \CYRS = {50,  },
⟨cmr⟩      \CYRT = {100, },
⟨pmn⟩      \CYRT = {70,  },
⟨cmr⟩      \CYRU = {100, },
⟨pmn⟩      \CYRU = {50,  },
⟨cmr⟩      \CYRF = {100, },
⟨cmr⟩      \CYRH = {50,  },
⟨cmr⟩      \CYRC = {50,  },
⟨cmr⟩      \CYRCH = {100, },
⟨cmr⟩      \CYRSH = {50,  },
⟨cmr⟩      \CYRSHCH = {50,  },
⟨cmr⟩      \CYRHRDSN = {100, },
⟨cmr⟩      \CYRERY = {50,  },
⟨cmr⟩      \CYRSFTSN = {50,  },
⟨cmr⟩      \CYREREV = {50,  },
⟨cmr⟩      \CYRYU = {50,  },
⟨cmr⟩      \CYRYA = {50,  },
⟨pmn⟩      \CYRYA = {  ,20},
```

```
⟨pmn⟩      \cyrr = {-50,  },
⟨m-t | pmn⟩       _ = {  ,100},
⟨cmr⟩       _ = {100,200},
⟨pmn⟩       031 = { ,-100}, % ffl
⟨pmn⟩      \v t = { ,100},
⟨m-t⟩      \textbackslash    = {100,200},   \quotedblbase      = {400,500},
⟨cmr⟩      \textbackslash    = {300,300},   \quotedblbase      = {200,600},
⟨pmn⟩      \textbackslash    = {100,150},   \quotedblbase      = {150,500},
⟨m-t⟩      \guillemotleft    = {300,300},   \guillemotright    = {300,300},
⟨cmr⟩      \guillemotleft    = {400,100},   \guillemotright    = {200,300},
⟨pmn⟩      \guillemotleft    = {200,300},   \guillemotright    = {150,400},
⟨m-t⟩      \textbraceleft    = {200,100},   \textbraceright    = {200,200},
⟨cmr⟩      \textbraceleft    = {400,100},   \textbraceright    = {200,200},
⟨pmn⟩      \textbraceleft    = {200,   },   \textbraceright    = {   ,200},
⟨cmr⟩      \textquotedblleft = {500,300},
⟨cmr⟩      \textless         = {300,100},   \textgreater       = {200,100}
⟨pmn⟩      \textless         = {100,   },   \textgreater       = {   ,100}
  }

⟨/m-t | cmr | pmn⟩
⟨*m-t | ptm⟩
\SetProtrusion
⟨m-t⟩   [ name     = QX-it-default,
⟨ptm⟩   [ name     = ptm-it-QX,
⟨m-t⟩     load     = OT1-it ]
⟨ptm⟩     load     = ptm-it ]
   { encoding = {QX},
⟨ptm⟩     family   = {ptm,ptmx,ptmj},
     shape    = {it,sl}  }
   {
⟨ptm⟩     009 = {  , 50}, % fk
     {=} = {100,100},
⟨m-t⟩     \textunderscore   = {100,100},
⟨ptm⟩     \textunderscore   = {100,150},
     \textbackslash    = {100,200},
     \quotedblbase     = {300,400},
⟨m-t⟩     \guillemotleft    = {300,300},   \guillemotright    = {300,300},
⟨ptm⟩     \guillemotleft    = {200,400},   \guillemotright    = {200,400},
     \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
     \textbraceleft    = {200,100},   \textbraceright    = {200,200},
     \textless         = {100,100},   \textgreater       = {100,100},
     \textminus        = {200,200},   \textdegree        = {300,150},
⟨m-t⟩     \copyright        = {100,100},   \textregistered    = {100,100}
⟨ptm⟩     \textregistered   = {100,150},   \copyright         = {100,150},
⟨ptm⟩     \textDelta        = { 70,   },   \textdelta         = {   , 50},
⟨ptm⟩     \textpi           = { 50, 80},   \textmu            = {   , 80},
⟨ptm⟩     \texteuro         = {200,   },   \textellipsis      = {100,200},
⟨ptm⟩     \textquoteleft    = {500,400},   \textquoteright    = {500,400},
⟨ptm⟩     \textquotedblleft = {500,300},   \textquotedblright = {400,400},
⟨ptm⟩     \textapprox       = { 50, 50},   \textinfty         = {100,100},
⟨ptm⟩     \textdagger       = {150,150},   \textdaggerdbl     = {100,100},
⟨ptm⟩     \textdiv          = {150,150},   \textasciitilde    = { 80, 80},
⟨ptm⟩     \texttimes        = {100,150},   \textpm            = { 50, 80},
⟨ptm⟩     \textbullet       = {300,100},   \textperiodcentered = {300,300},
⟨ptm⟩     \textquotesingle  = {500,500},   \textquotedbl      = {300,300},
⟨ptm⟩     \textperthousand  = {    ,50}
   }

⟨/m-t | ptm⟩
⟨*cmr | bch⟩
```

```
\SetProtrusion
⟨cmr⟩   [ name = cmr-it-T5,
⟨cmr⟩     load = cmr-it ]
⟨bch⟩   [ name = bch-it-T5,
⟨bch⟩     load = bch-it ]
   { encoding = T5,
⟨bch⟩     family   = bch,
⟨cmr⟩     family   = cmr,
     shape    = it }
   {
⟨bch⟩      _ = {  ,100},
⟨cmr⟩      _ = {100,200},
⟨bch⟩     \textbackslash    = {150,150},
⟨cmr⟩     \textbackslash    = {300,300},
⟨bch⟩     \quotesinglbase   = {200,500},   \quotedblbase      = {150,500},
⟨cmr⟩     \quotesinglbase   = {300,700},   \quotedblbase      = {200,600},
⟨bch⟩     \guilsinglleft    = {300,400},   \guilsinglright    = {200,500},
⟨cmr⟩     \guilsinglleft    = {500,300},   \guilsinglright    = {400,400},
⟨bch⟩     \guillemotleft    = {200,300},   \guillemotright    = {150,400},
⟨cmr⟩     \guillemotleft    = {400,100},   \guillemotright    = {200,300},
⟨bch⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,200},
⟨cmr⟩     \textbraceleft    = {400,100},   \textbraceright    = {200,200},
⟨bch⟩     \textless         = {100,   },   \textgreater       = {   ,100}
⟨cmr⟩     \textless         = {300,100},   \textgreater       = {200,100}
  }

⟨/cmr | bch⟩
```

Slanted is very similar to italic.

```
⟨*cmr⟩
\SetProtrusion
   [ name     = cmr-sl,
     load     = cmr-it-OT1 ]
   { encoding = {OT1,OT4},
     family   = cmr,
     shape    = sl  }
   {
      L = {  ,50},
      f = { ,-50},
      - = {300, },
     \textendash = {400,  },  \textemdash = {300,  }
   }

\SetProtrusion
   [ name     = cmr-sl-T1,
     load     = cmr-it-T1 ]
   { encoding = {T1,LY1},
     family   = cmr,
     shape    = sl  }
   {
      L = {  ,50},
      f = { ,-50},
      - = {300, },
     \textendash = {400,  },  \textemdash = {300,  }
   }

\SetProtrusion
   [ name     = cmr-sl-T2A,
     load     = cmr-it-T2A ]
   { 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).

```
⟨*!(blg | ugm)⟩
\SetProtrusion
⟨m-t⟩    [ name     = OT1-sc,
⟨bch⟩    [ name     = bch-sc,
⟨cmr⟩    [ name     = cmr-sc-OT1,
⟨pad⟩    [ name     = pad-sc,
⟨pmn⟩    [ name     = pmnj-sc,
⟨ppl⟩    [ name     = ppl-sc,
⟨ptm⟩    [ name     = ptm-sc,
⟨m-t⟩      load     = default ]
⟨bch⟩      load     = bch-default ]
⟨cmr⟩      load     = cmr-OT1 ]
⟨pad⟩      load     = pad-default ]
⟨pmn⟩      load     = pmnj-default ]
⟨ppl⟩      load     = ppl-default ]
⟨ptm⟩      load     = ptm-default ]
⟨m-t | bch | pad | pmn⟩    { encoding = OT1,
⟨cmr | ppl | ptm⟩    { encoding = {OT1,OT4},
⟨bch⟩      family   = bch,
⟨cmr⟩      family   = cmr,
⟨pad⟩      family   = {pad,padx,padj},
⟨pmn⟩      family   = pmnj,
⟨ppl⟩      family   = {ppl,pplx,pplj},
⟨ptm⟩      family   = {ptm,ptmx,ptmj},
      shape    = sc }
    {
      a = {50,50},
⟨cmr | pad | ppl | ptm⟩    \ae = {50,  },
⟨bch | pmn⟩      c = {50,  },
⟨bch | pad | pmn⟩      d = {  ,50},
⟨m-t | bch | cmr | pad | pmn | ptm⟩      f = {  ,50},
⟨bch | pad | pmn⟩      g = {50,  },
⟨m-t | cmr | pad | pmn | ppl | ptm⟩      j = {50,  },
⟨bch⟩      j = {100,  },
⟨m-t | bch | cmr | pad | pmn | ppl⟩      l = {  ,50},
⟨ptm⟩      l = {  ,80},
⟨m-t | bch | cmr | pad | pmn | ppl⟩    013 = {  ,50}, % fl
⟨ptm⟩    013 = {  ,80}, % fl
⟨bch | pad | pmn⟩      o = {50,50},
⟨pad | pmn⟩    \oe = {50,  },
⟨ppl⟩      p = { 0, 0},
⟨bch | pad | pmn⟩      q = {50,70},
⟨ppl⟩      q = { 0,  },
⟨m-t | cmr | pad | pmn | ppl | ptm⟩      r = {  , 0},
      t = {50,50},
⟨m-t | bch | cmr | pad | pmn | ppl⟩      y = {50,50}
⟨ptm⟩      y = {80,80}
    }

\SetProtrusion
⟨m-t⟩    [ name     = T1-sc,
⟨bch⟩    [ name     = bch-sc-T1,
⟨cmr⟩    [ name     = cmr-sc-T1,
⟨pad⟩    [ name     = pad-sc-T1,
```

```
⟨pmn⟩    [ name     = pmnj-sc-T1,
⟨ppl⟩    [ name     = ppl-sc-T1,
⟨ptm⟩    [ name     = ptm-sc-T1,
⟨m-t⟩      load     = T1-default ]
⟨bch⟩      load     = bch-T1     ]
⟨cmr⟩      load     = cmr-T1     ]
⟨pad⟩      load     = pad-T1     ]
⟨pmn⟩      load     = pmnj-T1    ]
⟨ppl⟩      load     = ppl-T1     ]
⟨ptm⟩      load     = ptm-T1     ]
   { encoding = {T1,LY1},
⟨bch⟩      family   = bch,
⟨cmr⟩      family   = cmr,
⟨pad⟩      family   = {pad,padx,padj},
⟨pmn⟩      family   = pmnj,
⟨ppl⟩      family   = {ppl,pplx,pplj},
⟨ptm⟩      family   = {ptm,ptmx,ptmj},
      shape    = sc }
   {
      a = {50,50},
⟨cmr | pad | ppl | ptm⟩    \ae = {50,  },
⟨bch | pmn⟩      c = {50,  },
⟨bch | pad | pmn⟩      d = {  ,50},
⟨m-t | bch | cmr | pad | pmn | ptm⟩      f = {  ,50},
⟨bch | pad | pmn⟩      g = {50,  },
⟨m-t | cmr | pad | pmn | ppl | ptm⟩      j = {50,  },
⟨bch⟩      j = {100,  },
⟨m-t | bch | cmr | pad | pmn | ppl⟩      l = {  ,50},
⟨ptm⟩      l = {  ,80},
⟨m-t | bch | cmr | pad | pmn | ppl⟩    029 = {  ,50}, % fl
⟨ptm⟩    029 = {  ,80}, % fl
⟨bch | pad | pmn⟩      o = {50,50},
⟨bch | pad | pmn⟩    \oe = {50,  },
⟨ppl⟩      p = { 0, 0},
⟨bch | pad | pmn⟩      q = {50,70},
⟨ppl⟩      q = { 0,  },
⟨m-t | cmr | pad | pmn | ppl | ptm⟩      r = {  , 0},
      t = {50,50},
⟨m-t | bch | cmr | pad | pmn | ppl⟩      y = {50,50}
⟨ptm⟩      y = {80,80}
   }

⟨/!(blg | ugm)⟩
⟨*m-t | cmr⟩
\SetProtrusion
⟨m-t⟩    [ name     = T2A-sc,
⟨cmr⟩    [ name     = cmr-sc-T2A,
⟨m-t⟩      load     = T2A-default ]
⟨cmr⟩      load     = cmr-T2A     ]
   { encoding = T2A,
⟨cmr⟩      family   = cmr,
      shape    = sc }
   {
      \cyra = {50,50},
      \cyrg = {  ,50},
      \cyrt = {50,50},
      \cyry = {  ,50}
   }

⟨/m-t | cmr⟩
```

```
⟨*m-t⟩
\SetProtrusion
   [ name     = QX-sc,
     load     = QX-default ]
   { encoding = QX,
     shape    = sc  }
   {
     a = {50,50},
     f = {  ,50},
     j = {50,  },
     l = {  ,50},
   013 = {  ,50}, % fl
     r = {  , 0},
     t = {50,50},
     y = {50,50}
   }

⟨/m-t⟩
⟨*cmr | bch⟩
\SetProtrusion
⟨bch⟩   [ name     = bch-sc-T5,
⟨bch⟩     load     = bch-T5 ]
⟨cmr⟩   [ name     = cmr-sc-T5,
⟨cmr⟩     load     = cmr-T5 ]
   { encoding = T5,
⟨bch⟩     family   = bch,
⟨cmr⟩     family   = cmr,
     shape    = sc }
   {
     a = {50,50},
⟨bch⟩     c = {50,  },
⟨bch⟩     d = {  ,50},
     f = {  ,50},
⟨bch⟩     g = {50,  },
⟨bch⟩     j = {100, },
⟨cmr⟩     j = {50,  },
     l = {  ,50},
⟨bch⟩     o = {50,50},
⟨bch⟩     q = { 0,  },
⟨cmr⟩     r = {  , 0},
     t = {50,50},
     y = {50,50}
   }

⟨/cmr | bch⟩
⟨*pmn⟩
\SetProtrusion
   [ name     = pmnx-sc,
     load     = pmnj-sc ]
   { encoding = OT1,
     family   = pmnx,
     shape    = sc }
   {
     1 = {230,180}
   }

\SetProtrusion
   [ name     = pmnx-sc-T1,
     load     = pmnj-sc-T1 ]
   { 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},
```

```
    b = {20,-50},
    c = {50,-50},
    d = {20, 0},
    e = {20,-50},
    f = {10, 0},
  028 = {10,-50}, % fi
  029 = {10,-50}, % fl
  030 = {10,-50}, % ffi
  031 = {10,-50}, % ffl
    g = {50,-50},
    i = {20,-50},
  188 = {20, 0},  % ij
    j = {20, 0},
    k = {20,  },
    l = {20,50},
    m = {  ,-30},
    n = {  ,-30},
    o = {50,  },
  \oe = {50,-50},
    p = {20,-50},
    q = {50,  },
    r = {20, 0},
    s = {20,-30},
    t = {70,  },
    u = {50,-50},
    v = {100,  },
    w = {100,  },
    y = {50,  },
    z = {  ,-50}
  }

\SetProtrusion
  [ name     = pmnx-scit,
    load     = pmnj-scit ]
  { encoding = OT1,
    family   = pmnx,
    shape    = {scit,si} }
  {
    1 = {100,150}
  }

\SetProtrusion
  [ name     = pmnx-scit-T1,
    load     = pmnj-scit-T1 ]
  { encoding = {T1,LY1},
    family   = pmnx,
    shape    = {scit,si}    }
  {
    1 = {100,150}
  }

⟨/pmn⟩
```

15.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```
\SetProtrusion
⟨m-t⟩   [ name     = textcomp ]
```

```
⟨bch⟩    [ name     = bch-textcomp ]
⟨blg⟩    [ name     = blg-textcomp ]
⟨cmr⟩    [ name     = cmr-textcomp ]
⟨pad⟩    [ name     = pad-textcomp ]
⟨pmn⟩    [ name     = pmn-textcomp ]
⟨ppl⟩    [ name     = ppl-textcomp ]
⟨ptm⟩    [ name     = ptm-textcomp ]
⟨ugm⟩    [ name     = ugm-textcomp ]
⟨m-t⟩    { encoding = TS1      }
⟨!m-t⟩   { encoding = TS1,
⟨bch⟩      family   = bch }
⟨blg⟩      family   = blg }
⟨cmr⟩      family   = cmr }
⟨pad⟩      family   = {pad,padx,padj} }
⟨pmn⟩      family   = {pmnx,pmnj} }
⟨ppl⟩      family   = {ppl,pplx,pplj} }
⟨ptm⟩      family   = {ptm,ptmx,ptmj} }
⟨ugm⟩      family   = ugm }
    {
⟨blg⟩      \textquotestraightbase    = {400,500},
⟨cmr⟩      \textquotestraightbase    = {300,300},
⟨pad | pmn⟩      \textquotestraightbase    = {400,400},
⟨blg⟩      \textquotestraightdblbase = {300,400},
⟨cmr | pmn⟩      \textquotestraightdblbase = {300,300},
⟨pad⟩      \textquotestraightdblbase = {400,400},
⟨bch | cmr | pad | pmn | ugm⟩      \texttwelveudash          = {200,200},
⟨bch | cmr | pad | pmn⟩      \textthreequartersemdash  = {150,150},
⟨ugm⟩      \textthreequartersemdash  = {200,200},
⟨blg⟩      \textquotesingle          = {500,600},
⟨cmr | pmn⟩      \textquotesingle          = {300,400},
⟨pad⟩      \textquotesingle          = {400,500},
⟨ptm⟩      \textquotesingle          = {500,500},
⟨ugm⟩      \textquotesingle          = {300,500},
⟨bch | cmr | pmn⟩      \textasteriskcentered     = {200,300},
⟨blg⟩      \textasteriskcentered     = {150,200},
⟨pad⟩      \textasteriskcentered     = {300,300},
⟨ugm⟩      \textasteriskcentered     = {100,200},
⟨pmn⟩      \textfractionsolidus      = {-200,-200},
⟨cmr⟩      \textoneoldstyle          = {100,100},
⟨pmn⟩      \textoneoldstyle          = {   , 50},
⟨cmr⟩      \textthreeoldstyle        = {   , 50},
⟨pad | pmn⟩      \textthreeoldstyle        = { 50,   },
⟨cmr⟩      \textfouroldstyle         = { 50, 50},
⟨pad | pmn⟩      \textfouroldstyle         = { 50,   },
⟨cmr | pad | pmn⟩      \textsevenoldstyle        = { 50, 80},
⟨cmr⟩      \textlangle               = {400,   },
⟨cmr⟩      \textrangle               = {   ,400},
⟨m-t | bch | pmn | ptm⟩      \textminus                = {200,200},
⟨cmr | pad | ppl⟩      \textminus                = {300,300},
⟨blg | ugm⟩      \textminus                = {250,300},
⟨bch | pad | pmn⟩      \textlbrackdbl            = {100,   },
⟨blg⟩      \textlbrackdbl            = {200,   },
⟨bch | pad | pmn⟩      \textrbrackdbl            = {   ,100},
⟨blg⟩      \textrbrackdbl            = {   ,200},
⟨pmn⟩      \textasciigrave           = {200,500},
⟨bch | blg | cmr | pad | pmn⟩      \texttildelow             = {200,250},
⟨pmn⟩      \textasciibreve           = {300,400},
⟨pmn⟩      \textasciicaron           = {300,400},
⟨pmn⟩      \textacutedbl             = {200,300},
⟨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},
```

```
⟨cmr | pad | pmn⟩     \textthreequartersemdash  = {150,150},
⟨ugm⟩      \textthreequartersemdash  = {200,200},
⟨cmr⟩      \textquotesingle          = {600,300},
⟨pad⟩      \textquotesingle          = {800,100},
⟨pmn⟩      \textquotesingle          = {300,200},
⟨ugm⟩      \textquotesingle          = {500,500},
⟨cmr⟩      \textasteriskcentered     = {300,200},
⟨pad⟩      \textasteriskcentered     = {500,100},
⟨pmn⟩      \textasteriskcentered     = {200,300},
⟨ugm⟩      \textasteriskcentered     = {300,150},
⟨pmn⟩      \textfractionsolidus      = {-200,-200},
⟨cmr⟩      \textoneoldstyle          = {100, 50},
⟨pad⟩      \textoneoldstyle          = {100,   },
⟨pmn⟩      \textoneoldstyle          = { 50,   },
⟨pad⟩      \texttwooldstyle          = { 50,   },
⟨pmn⟩      \texttwooldstyle          = {-50,   },
⟨cmr⟩      \textthreeoldstyle        = {100, 50},
⟨pmn⟩      \textthreeoldstyle        = {-100,  },
⟨cmr⟩      \textfouroldstyle         = { 50, 50},
⟨pad⟩      \textfouroldstyle         = { 50,100},
⟨cmr⟩      \textsevenoldstyle        = { 50, 80},
⟨pad⟩      \textsevenoldstyle        = { 50,   },
⟨pmn⟩      \textsevenoldstyle        = { 20,   },
⟨cmr⟩      \textlangle               = {400,   },
⟨cmr⟩      \textrangle               = {   ,400},
⟨cmr | pad⟩     \textminus               = {300,300},
⟨pmn⟩      \textminus                = {200,200},
⟨ugm⟩      \textminus                = {250,300},
⟨pad | pmn⟩     \textlbrackdbl           = {100,   },
⟨pad | pmn⟩     \textrbrackdbl           = {   ,100},
⟨pmn⟩      \textasciigrave           = {300,300},
⟨cmr | pad | pmn⟩     \texttildelow            = {200,250},
⟨pmn⟩      \textasciibreve           = {300,300},
⟨pmn⟩      \textasciicaron           = {300,300},
⟨pmn⟩      \textacutedbl             = {200,300},
⟨pmn⟩      \textgravedbl             = {150,300},
⟨cmr⟩      \textdagger               = {100,100},
⟨pad⟩      \textdagger               = {200,100},
⟨pmn⟩      \textdagger               = { 80, 50},
⟨ugm⟩      \textdagger               = { 80, 80},
⟨cmr | pad⟩     \textdaggerdbl           = { 80, 80},
⟨pmn⟩      \textdaggerdbl            = { 80, 50},
⟨ugm⟩      \textbardbl               = {150,150},
⟨cmr⟩      \textbullet               = {200,100},
⟨pad⟩      \textbullet               = {300,   },
⟨pmn⟩      \textbullet               = { 30, 70},
⟨ugm⟩      \textbullet               = { 50,100},
⟨cmr⟩      \textcelsius              = {100,   },
⟨pad⟩      \textcelsius              = {200,   },
⟨pmn⟩      \textcelsius              = { 50,-50},
⟨pad⟩      \textflorin               = {100,   },
⟨pmn⟩      \textflorin               = { 50,100},
⟨ugm⟩      \textflorin               = {   ,100},
⟨cmr⟩      \textcolonmonetary        = {150,   },
⟨pad⟩      \textcolonmonetary        = {100,   },
⟨pmn⟩      \textcolonmonetary        = { 50,-50},
⟨cmr | pad⟩     \texttrademark           = {200,   },
⟨pmn⟩      \texttrademark            = { 50,100},
⟨ugm⟩      \texttrademark            = {150, 50},
⟨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   }
```

⟨/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}
```

⟨*cmr⟩

```
\SetProtrusion
   [ name     = cmr-math-letters ]
   { encoding = OML,
     family   = cmm,
     series   = {m,b},
     shape    = it    }
   {
       A = {100, 50}, % \mathnormal
       B = { 50,   },
       C = { 50,   },
       D = { 50, 50},
       E = { 50,   },
       F = {100, 50},
       G = { 50, 50},
       H = { 50, 50},
       I = { 50, 50},
       J = {150, 50},
       K = { 50,100},
       L = { 50, 50},
       M = { 50,   },
       N = { 50,   },
       O = { 50,   },
       P = { 50,   },
       Q = { 50, 50},
       R = { 50,   },
       S = { 50,   },
       T = { 50,100},
       U = { 50, 50},
       V = {100,100},
       W = { 50,100},
       X = { 50,100},
       Y = {100,100},
       f = {100,100},
       h = {   ,100},
       i = {   , 50},
       j = {   , 50},
```

```
      k = {   , 50},
      r = {   , 50},
      v = {   , 50},
      w = {   , 50},
      x = {   , 50},
    "0B = { 50,100}, % \alpha
    "0C = { 50, 50}, % \beta
    "0D = {200,150}, % \gamma
    "0E = { 50, 50}, % \delta
    "0F = { 50, 50}, % \epsilon
    "10 = { 50,150}, % \zeta
    "12 = { 50,   }, % \theta
    "13 = {   ,100}, % \iota
    "14 = {   ,100}, % \kappa
    "15 = {100, 50}, % \lambda
    "16 = {   , 50}, % \mu
    "17 = {   , 50}, % \nu
    "18 = {   , 50}, % \xi
    "19 = { 50,100}, % \pi
    "1A = { 50, 50}, % \rho
    "1B = {   ,150}, % \sigma
    "1C = { 50,150}, % \tau
    "1D = { 50, 50}, % \upsilon
    "1F = { 50,100}, % \chi
    "20 = { 50, 50}, % \psi
    "21 = {   , 50}, % \omega
    "22 = {   , 50}, % \varepsilon
    "23 = {   , 50}, % \vartheta
    "24 = {   , 50}, % \varpi
    "25 = {100,   }, % \varrho
    "26 = {100,100}, % \varsigma
    "27 = { 50, 50}, % \varphi
    "28 = {100,100}, % \leftharpoonup
    "29 = {100,100}, % \leftharpoondown
    "2A = {100,100}, % \rightharpoonup
    "2B = {100,100}, % \rightharpoondown
    "2C = {300,200}, % \lhook
    "2D = {200,300}, % \rhook
    "2E = {   ,100}, % \triangleright
    "2F = {100,   }, % \triangleleft
    "3A = {   ,500}, % ., \ldotp
    "3B = {   ,500}, % ,
    "3C = {200,100}, % <
    "3D = {300,400}, % /
    "3E = {100,200}, % >
    "3F = {200,200}, % \star
    "5B = {   ,100}, % \flat
    "5E = {200,200}, % \smile
    "5F = {200,200}, % \frown
    "7C = {100,   }, % \jmath
    "7D = {   ,100}  % \wp
```

Remaining slots in the source file.

```
  }

```

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}
```

```
\SetProtrusion
   [ name     = cmr-math-symbols ]
   { encoding = OMS,
     family   = cmsy,
     series   = {m,b},
     shape    = n  }
   {
       A = {150, 50}, % \mathcal
       C = {   ,100},
       D = {   , 50},
       F = { 50,150},
       I = {   ,100},
       J = {100,150},
       K = {   ,100},
       L = {100,   },
       M = { 50, 50},
       N = { 50,100},
       P = {   , 50},
       Q = { 50,   },
       R = {   , 50},
       T = { 50,150},
       V = { 50, 50},
       W = {   , 50},
       X = {100,100},
       Y = {100,   },
       Z = {100,150},
     "00 = {300,300}, % -
     "01 = {   ,700}, % \cdot, \cdotp
     "02 = {150,250}, % \times
     "03 = {150,250}, % *, \ast
     "04 = {200,300}, % \div
     "05 = {150,250}, % \diamond
     "06 = {200,200}, % \pm
     "07 = {200,200}, % \mp
     "08 = {100,100}, % \oplus
     "09 = {100,100}, % \ominus
     "0A = {100,100}, % \otimes
     "0B = {100,100}, % \oslash
     "0C = {100,100}, % \odot
     "0D = {100,100}, % \bigcirc
     "0E = {100,100}, % \circ
     "0F = {100,100}, % \bullet
     "10 = {100,100}, % \asymp
     "11 = {100,100}, % \equiv
     "12 = {200,100}, % \subseteq
     "13 = {100,200}, % \supseteq
     "14 = {200,100}, % \leq
     "15 = {100,200}, % \geq
     "16 = {200,100}, % \preceq
     "17 = {100,200}, % \succeq
     "18 = {200,200}, % \sim
     "19 = {150,150}, % \approx
     "1A = {200,100}, % \subset
     "1B = {100,200}, % \supset
     "1C = {200,100}, % \ll
     "1D = {100,200}, % \gg
     "1E = {300,100}, % \prec
     "1F = {100,300}, % \succ
     "20 = {100,200}, % \leftarrow
     "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).

⟨∗cfg-u⟩

Symbol font 'a'.

```
⟨∗msa⟩
\SetProtrusion
   [ name     = AMS-a ]
   { encoding = U,
     family   = msa  }
   {
     "05  =  {150,250},  % \centerdot
     "06  =  {100,100},  % \lozenge
     "07  =  { 50, 50},  % \blacklozenge
     "08  =  { 50, 50},  % \circlearrowright
     "09  =  { 50, 50},  % \circlearrowleft
     "0A  =  {100,100},  % \rightleftharpoons
     "0B  =  {100,100},  % \leftrightharpoons
     "0D  =  {-50,200},  % \Vdash
     "0E  =  {-50,200},  % \Vvdash
     "0F  =  {-70,150},  % \vDash
     "10  =  {100,150},  % \twoheadrightarrow
     "11  =  {100,150},  % \twoheadleftarrow
     "12  =  { 50,100},  % \leftleftarrows
     "13  =  { 50, 80},  % \rightrightarrows
     "14  =  {120,120},  % \upuparrows
     "15  =  {120,120},  % \downdownarrows
     "16  =  {200,200},  % \upharpoonright
     "17  =  {200,200},  % \downharpoonright
     "18  =  {200,200},  % \upharpoonleft
     "19  =  {200,200},  % \downharpoonleft
     "1A  =  { 80,100},  % \rightarrowtail
     "1B  =  { 80,100},  % \leftarrowtail
     "1C  =  { 50, 50},  % \leftrightarrows
     "1D  =  { 50, 50},  % \rightleftarrows
     "1E  =  {250,   },  % \Lsh
     "1F  =  {   ,250},  % \Rsh
     "20  =  {100,100},  % \rightsquigarrow
     "21  =  {100,100},  % \leftrightsquigarrow
     "22  =  {100, 50},  % \looparrowleft
     "23  =  { 50,100},  % \looparrowright
     "24  =  { 50, 80},  % \circeq
     "25  =  {   ,100},  % \succsim
     "26  =  {   ,100},  % \gtrsim
     "27  =  {   ,100},  % \gtrapprox
     "28  =  {150, 50},  % \multimap
     "2B  =  {100,150},  % \doteqdot
     "2C  =  {100,150},  % \triangleq
     "2D  =  {100, 50},  % \precsim
     "2E  =  {100, 50},  % \lesssim
     "2F  =  { 50, 50},  % \lessapprox
     "30  =  {100, 50},  % \eqslantless
     "31  =  { 50, 50},  % \eqslantgtr
     "32  =  {100, 50},  % \curlyeqprec
     "33  =  { 50,100},  % \curlyeqsucc
     "34  =  {100, 50},  % \preccurlyeq
     "36  =  { 50,   },  % \leqslant
     "38  =  {   , 50},  % \backprime
     "39  =  {250,250},  % \dabar@ : the dash bar in \dash(left,right)arrow
```

```
    "3C  =  { 50,100},  % \succcurlyeq
    "3E  =  {   , 50},  % \geqslant
    "40  =  {   , 50},  % \sqsubset
    "41  =  { 50,   },  % \sqsupset
    "42  =  {   ,150},  % \vartriangleright, \rhd
    "43  =  {150,   },  % \vartriangleleft, \lhd
    "44  =  {   ,100},  % \trianglerighteq, \unrhd
    "45  =  {100,   },  % \triangleleftteq, \unlhd
    "46  =  {100,100},  % \bigstar
    "48  =  { 50, 50},  % \blacktriangledown
    "49  =  {   ,100},  % \blacktriangleright
    "4A  =  {100,   },  % \blacktriangleleft
    "4B  =  {   ,150},  % \dashrightarrow (the arrow)
    "4C  =  {150,   },  % \dashleftarrow
    "4D  =  { 50, 50},  % \vartriangle
    "4E  =  { 50, 50},  % \blacktriangle
    "4F  =  { 50, 50},  % \triangledown
    "50  =  { 50, 50},  % \eqcirc
    "56  =  {   ,150},  % \Rrightarrow
    "57  =  {150,   },  % \Lleftarrow
    "58  =  {100,300},  % \checkmark
    "5C  =  { 50, 50},  % \angle
    "5D  =  { 50, 50},  % \measuredangle
    "5E  =  { 50, 50},  % \sphericalangle
    "5F  =  {   , 50},  % \varpropto
    "60  =  {100,100},  % \smallsmile
    "61  =  {100,100},  % \smallfrown
    "62  =  { 50,   },  % \Subset
    "63  =  {   , 50},  % \Supset
    "66  =  {150,150},  % \curlywedge
    "67  =  {150,150},  % \curlyvee
    "68  =  { 50,150},  % \leftthreetimes
    "69  =  {100, 50},  % \rightthreetimes
    "6C  =  { 50, 50},  % \bumpeq
    "6D  =  { 50, 50},  % \Bumpeq
    "6E  =  {100,   },  % \lll
    "6F  =  {   ,100},  % \ggg
    "70  =  { 50,100},  % \ulcorner
    "71  =  {100, 50},  % \urcorner
    "75  =  {150,200},  % \dotplus
    "76  =  { 50,100},  % \backsim
    "78  =  { 50,100},  % \llcorner
    "79  =  {100, 50},  % \lrcorner
    "7C  =  {100,100},  % \intercal
    "7D  =  { 50, 50},  % \circledcirc
    "7E  =  { 50, 50},  % \circledast
    "7F  =  { 50, 50}   % \circleddash
```

Remaining slots in the source file.

```
  }

⟨/msa⟩
```

Symbol font ‘b’.

```
⟨*msb⟩
\SetProtrusion
  [ name     = AMS-b ]
  { encoding = U,
    family   = msb  }
  {
```

```
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
```

```
      "32  =  {   ,100},  % \nvDash
      "33  =  {   ,100},  % \nVDash
      "34  =  {   ,100},  % \ntrianglerighteq
      "35  =  {100,   },  % \ntrianglelefteq
      "36  =  {100,   },  % \ntriangleleft
      "37  =  {   ,100},  % \ntriangleright
      "38  =  {100,200},  % \nleftarrow
      "39  =  {100,200},  % \nrightarrow
      "3A  =  {100,100},  % \nLeftarrow
      "3B  =  { 50,100},  % \nRightarrow
      "3C  =  {100,100},  % \nLeftrightarrow
      "3D  =  {100,200},  % \nleftrightarrow
      "3E  =  { 50, 50},  % \divideontimes
      "3F  =  { 50, 50},  % \varnothing
      "60  =  {200,   },  % \Finv
      "61  =  {   , 50},  % \Game
      "68  =  {100,100},  % \eqsim
      "69  =  { 50,   },  % \beth
      "6A  =  { 50,   },  % \gimel
      "6B  =  {150,   },  % \daleth
      "6C  =  {200,   },  % \lessdot
      "6D  =  {   ,200},  % \gtrdot
      "6E  =  {100,200},  % \ltimes
      "6F  =  {150,100},  % \rtimes
      "70  =  { 50,100},  % \shortmid
      "71  =  { 50, 50},  % \shortparallel
      "72  =  {200,300},  % \smallsetminus
      "73  =  {100,200},  % \thicksim
      "74  =  { 50,100},  % \thickapprox
      "75  =  { 50, 50},  % \approxeq
      "76  =  { 50,100},  % \succapprox
      "77  =  { 50, 50},  % \precapprox
      "78  =  {100,100},  % \curvearrowleft
      "79  =  { 50,150},  % \curvearrowright
      "7A  =  { 50,200},  % \digamma
      "7B  =  {100, 50},  % \varkappa
      "7F  =  {200,   }   % \backepsilon
```

Remaining slots in the source file.

```
   }

⟨/msb⟩
```

15.8.8 Euler

Euler Roman font (package euler).

```
⟨*eur⟩
\SetProtrusion
   [ name     = euler ]
   { encoding = U,
     family   = eur  }
   {
     "01  =  {100,100},
     "03  =  {100,150},
     "06  =  {   ,100},
     "07  =  {100,150},
     "08  =  {100,100},
     "0A  =  {100,100},
     "0B  =  {   , 50},
```

```
    "0C  =  {   ,100},
    "0D  =  {100,100},
    "0E  =  {   ,100},
    "0F  =  {100,100},
    "10  =  {100,100},
    "13  =  {   ,100},
    "14  =  {   ,100},
    "15  =  {   , 50},
    "16  =  {   , 50},
    "17  =  { 50,100},
    "18  =  { 50,100},
    "1A  =  {   , 50},
    "1B  =  {   , 50},
    "1C  =  { 50,100},
    "1D  =  { 50,100},
    "1E  =  { 50,100},
    "1F  =  { 50,100},
    "20  =  {   , 50},
    "21  =  {   , 50},
    "22  =  { 50,100},
    "24  =  {   , 50},
    "27  =  { 50,100},
      1  =  {100,100},
      7  =  { 50,100},
    "3A  =  {300,500},
    "3B  =  {200,400},
    "3C  =  {200,100},
    "3D  =  {200,200},
    "3E  =  {100,200},
      A  =  {   ,100},
      D  =  {   , 50},
      J  =  { 50,   },
      K  =  {   , 50},
      L  =  {   , 50},
      Q  =  {   , 50},
      T  =  { 50,   },
      X  =  { 50, 50},
      Y  =  { 50,   },
      h  =  {   , 50},
      k  =  {   , 50}
  }

```

Extended by the eulervm package.

```
\SetProtrusion
  [ name     = euler-vm,
    load     = euler ]
  { encoding = U,
    family   = zeur  }
  {
    "28  =  {100,200},
    "29  =  {100,200},
    "2A  =  {100,150},
    "2B  =  {100,150},
    "2C  =  {200,300},
    "2D  =  {200,300},
    "2E  =  {   ,100},
    "2F  =  {100,   },
    "3F  =  {150,150},
    "5B  =  {   ,100},
```

```
    "5E  =  {100,100},
    "5F  =  {100,100},
    "80  =  {   , 50},
    "81  =  {200,250},
    "82  =  {100,200}
  }

⟨/eur⟩
```

Euler Script font (eucal).

```
⟨*eus⟩
\SetProtrusion
  [ name     = euscript ]
  { encoding = U,
    family   = eus  }
  {
      A  =  {100,100},
      B  =  { 50,100},
      C  =  { 50, 50},
      D  =  { 50,100},
      E  =  { 50,100},
      F  =  { 50,   },
      G  =  { 50,   },
      H  =  {   ,100},
      K  =  {   , 50},
      L  =  {   ,150},
      M  =  {   , 50},
      N  =  {   , 50},
      O  =  { 50, 50},
      P  =  { 50, 50},
      T  =  {   ,100},
      U  =  {   , 50},
      V  =  { 50, 50},
      W  =  { 50, 50},
      X  =  { 50, 50},
      Y  =  { 50,   },
      Z  =  { 50,100},
    "00  =  {250,250},
    "18  =  {200,200},
    "3A  =  {200,150},
    "40  =  {   ,100},
    "5E  =  {100,100},
    "5F  =  {100,100},
    "66  =  { 50,   },
    "67  =  {   , 50},
    "6E  =  {200,200}
  }

\SetProtrusion
  [ name     = euscript-vm,
    load     = euscript ]
  { encoding = U,
    family   = zeus  }
  {
    "01  =  {600,600},
    "02  =  {200,200},
    "03  =  {200,200},
    "04  =  {200,200},
    "05  =  {150,150},
    "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},
```

```
      "73  =  {200,100},
      "76  =  {   ,100},
      "77  =  {100,   },
      "78  =  { 50, 50},
      "79  =  {100,100},
      "7A  =  {100,100},
      "7D  =  {150,150},
      "7E  =  {100,100},
      "A8  =  {100,100},
      "A9  =  {100,100},
      "AB  =  {200,200},
      "BA  =  {   ,200},
      "BB  =  {   ,200},
      "BD  =  {200,200},
      "DE  =  {200,200}
   }

⟨/eus⟩
```

Euler Fraktur font (eufrak).

```
⟨*euf⟩
\SetProtrusion
   [ name     = mathfrak ]
   { encoding = U,
     family   = euf  }
   {
       A  =  {   , 50},
       B  =  {   , 50},
       C  =  { 50, 50},
       D  =  {   , 80},
       E  =  { 50,   },
       G  =  {   , 50},
       L  =  {   , 80},
       O  =  {   , 50},
       T  =  {   , 80},
       X  =  { 80, 50},
       Z  =  { 80, 50},
       b  =  {   , 50},
       c  =  {   , 50},
       k  =  {   , 50},
       p  =  {   , 50},
       q  =  { 50,   },
       v  =  {   , 50},
       w  =  {   , 50},
       x  =  {   , 50},
       1  =  {100,100},
       2  =  { 80, 80},
       3  =  { 80, 50},
       4  =  { 80, 50},
       7  =  { 50, 50},
      "12  =  {500,500},
      "13  =  {500,500},
       !  =  {   ,200},
       ’  =  {200,300},
       (  =  {200,   },
       )  =  {   ,200},
       *  =  {200,200},
       +  =  {200,250},
       -  =  {200,200},
      {,} =  {300,300},
```

```
        .  =  {400,400},
      {=} =  {200,200},
       :  =  {   ,200},
       ;  =  {   ,200},
       ]  =  {   ,200}
   }

⟨/euf⟩
⟨/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]).

```
⟨*cfg-e⟩
\SetProtrusion
⟨zpeu | euroitc⟩   { encoding = U,
⟨mvs⟩   { encoding = {OT1,U},
⟨zpeu⟩     family   = zpeu }
⟨euroitc⟩     family   = {euroitc,euroitcs} }
⟨mvs⟩     family   = mvs }
   {
⟨zpeu⟩     E = {50, }
⟨euroitc⟩     E = {100,50}
⟨mvs⟩     164 = {50,50},   % \EUR
⟨mvs⟩     068 = {50,-100}  % \EURdig
   }

⟨*zpeu | euroitc⟩
\SetProtrusion
   { encoding = U,
⟨zpeu⟩     family   = zpeu,
⟨euroitc⟩     family   = {euroitc,euroitcs},
     shape    = it* }
   {
⟨zpeu⟩     E = {100,-50}
⟨euroitc⟩     E = {100,}
   }

⟨/zpeu | euroitc⟩
⟨*zpeu⟩
\SetProtrusion
   { encoding = U,
     family   = {zpeus,eurosans} }
   {
     E = {100,50}
   }

\SetProtrusion
   { encoding = U,
     family   = {zpeus,eurosans},
     shape    = it* }
   {
     E = {200, }
   }

⟨/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

```
        i  = { , 50, -50},
        m  = { , 50, -50},
        n  = { , 50, -50},
        u  = { , 50, -50},
```

- after colon and semicolon

```
        :  = { ,200,-200},
        ;  = { ,200,-200},
```

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

```
        .  = { ,250,-250},
        !  = { ,250,-250},
        ?  = { ,250,-250}
```

The order has to be reversed when enlarging is needed.'

```
  }
```

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]

```
⟨*m-t⟩
\SetExtraSpacing
   [ name     = T2A,
     load     = default ]
   { encoding = T2A,
     family   = cmr }
   {
      \cyrg  = { ,-300,300},
      \cyrb  = { ,-200,200},
      \cyrk  = { ,-200,200},
      \cyrs  = { ,-100,100},
      \cyrr  = { ,-100,100},
      \cyrh  = { ,-100,100},
      \cyru  = { ,-100,100},
      \cyrt  = { , 50, -50},
      \cyrp  = { , 50, -50},
```

23 Contributed by *Karl Karlsson*.

```
      \cyri  = { , 50, -50},
      \cyrishrt  = { , 50, -50},
   }

⟨/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.

```
\SetExtraSpacing
   [ name     = nonfrench-cmr,
     load     = default,
     context  = nonfrench ]
   { encoding = {OT1,T1,LY1,OT4,QX,T5},
     family   = cmr }
   {
```

`latex.ltx` has:

```
\def\nonfrenchspacing{
  \sfcode`\. 3000
  \sfcode`\? 3000
  \sfcode`\! 3000
```

```
     . = {333,2000,-667},
     ? = {333,2000,-667},
     ! = {333,2000,-667},
```

```
  \sfcode`\: 2000
```

```
     : = {333,1000,-500},
```

```
  \sfcode`\; 1500
```

```
     ; = {   , 500,-333},
```

```
  \sfcode`\, 1250
```

```
    {,}= {   , 250,-200}
```

```
}
```

```
   }

```

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.

```
\SetExtraSpacing
   [ name     = nonfrench-default,
     load     = default,
     context  = nonfrench ]
   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
   {
     . = {240,2000,-667},
     ? = {240,2000,-667},
     ! = {240,2000,-667},
     : = {240,1000,-500},
     ; = {   , 500,-333},
    {,}= {   , 250,-200}
   }

```

15.10 Additional kerning

Default unit is 1 em.

```
%%% ------------------------------------------------------------------------
%%% ADDITIONAL KERNING

```

A dummy list to be loaded when no context is active.

```
\SetExtraKerning
   [ name = empty ]
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
   { }

```

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.)

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

24 http://fr.wikipedia.org/wiki/Espace_typographique, 5 July 2007.

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfTEX.

```
\SetExtraKerning
   [ name     = french-guillemets,
     context  = french-guillemets,
     load     = french-default,
     unit     = space   ]
   { encoding = {T1,LY1} }
   {
    \guillemotleft  = { ,800}, % = 0.8\fontdimen2
    \guillemotright = {800, }
   }

\SetExtraKerning
   [ name     = french-guillemets-OT1,
     context  = french-guillemets,
     load     = french-default,
     unit     = space   ]
   { encoding = OT1     }
   { }
```

15.10.2 Turkish

```
\SetExtraKerning
   [ name     = turkish,
     context  = turkish ]
   { encoding = {OT1,T1,LY1} }
   {
     :  = {167, }, % = \thinspace
     !  = {167, },
    {=} = {167, }
   }
```

⟨/m-t⟩
⟨/config⟩

16 Auxiliary file for micro fine tuning

This file can be used to test protrusion and expansion settings.

⟨*test⟩

```
\documentclass{article}

%% Here you can specify the font you want to test, using
%% the commands \fontfamily, \fontseries and \fontshape.
%% Make sure to end all lines with a comment character!
\newcommand*\TestFont{%
  \fontfamily{ppl}%
%%  \fontseries{b}%
%%  \fontshape{it}% sc, sl
}

\usepackage{ifthen}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[verbose,expansion=alltext,stretch=50]{microtype}
```

```
\pagestyle{empty}
\setlength{\parindent}{0pt}
\newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
\newcommand*\testprotrusion[2][]{%
  \ifthenelse{\equal{#1}{r}}{}{#2}%
  lorem ipsum dolor sit amet,
    \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
    \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
  you know the rest%
  \ifthenelse{\equal{#1}{l}}{}{#2}%
  \linebreak
  {\fontencoding{\encodingdefault}%
  \fontseries{\seriesdefault}%
  \fontshape{\shapedefault}%
  \selectfont
  Here is the beginning of a line, \dotfill and here is its end}\linebreak
}
\newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
\def\stripprefix#1>{}
\newcount\charcount
\begin{document}

\microtypesetup{expansion=false}

{\centering The font in this document is called by:\\
 \texttt{\showTestFont}\par}\bigskip

\TestFont\selectfont
 This line intentionally left empty\linebreak
%% A -- Z
\charcount=65
\loop
  \testprotrusion{\char\charcount}
  \advance\charcount 1
  \ifnum\charcount < 91 \repeat
%% a -- z
\charcount=97
\loop
  \testprotrusion{\char\charcount}
  \advance\charcount 1
  \ifnum\charcount < 123 \repeat
%% 0 -- 9
\charcount=48
\loop
  \testprotrusion{\char\charcount}
  \advance\charcount 1
  \ifnum\charcount < 58 \repeat
%%
 \testprotrusion[r]{,}
 \testprotrusion[r]{.}
 \testprotrusion[r]{;}
 \testprotrusion[r]{:}
 \testprotrusion[r]{?}
 \testprotrusion[r]{!}
 \testprotrusion[l]{\textexclamdown}
 \testprotrusion[l]{\textquestiondown}
 \testprotrusion[r]{)}
 \testprotrusion[l]{(}
 \testprotrusion{/}
```

```
 \testprotrusion{\char`\\}
 \testprotrusion{-}
 \testprotrusion{\textendash}
 \testprotrusion{\textemdash}
 \testprotrusion{\textquoteleft}
 \testprotrusion{\textquoteright}
 \testprotrusion{\textquotedblleft}
 \testprotrusion{\textquotedblright}
 \testprotrusion{\quotesinglbase}
 \testprotrusion{\quotedblbase}
 \testprotrusion{\guilsinglleft}
 \testprotrusion{\guilsinglright}
 \testprotrusion{\guillemotleft}
 \testprotrusion{\guillemotright}

\newpage
The following displays the current font stretched by 5\%,
normal, and shrunk by 5\%:

\bigskip
\newlength{\MTln}
\newcommand*\teststring
  {ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
\settowidth{\MTln}{\teststring}
\microtypesetup{expansion=true}

\parbox{1.05\MTln}{\teststring\linebreak\\
                   \teststring}\par\bigskip
\parbox{0.95\MTln}{\teststring}

\end{document}
```
⟨/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

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.