Accessibility-related talks
Ross Moore, Macquarie University, Sydney
TUG 2020, Online: 25–27 July 2020
July 24-26, 2020

41st Annual Conference of the TeX Users Group

TeX and LaTeX Typography Typesetting Fonts Design Publishing and More
Accessibility, text-extraction
What is Accessibility?

What is involved in claiming an electronic document to be accessible?
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ extract characters correctly
What is Accessibility?

What is involved in claiming an electronic document to be *accessible*?

- extract characters correctly

- interword spaces

- e.g., for Figures, Formulas, Tables
  - not the same as a caption

- e.g., Window title.
  - Helps decide: is this the right le?
  - Do I want to read this?
  - How to find related information?

- Don't underestimate the value of Metadata.

- Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, ...

- PDF/A-#, PDF/UA, `Matterhorn Protocol'
  - Acrobat Pro's Accessibility tests
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ extract characters correctly

▶ interword spaces

▶ soft semantics

▶ alternative text
e.g., for Figures, Formulas, Tables

▶ Metadata
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- extract characters correctly
- interword spaces
- soft semantics
- alternative text
  e.g., for Figures, Formulas, Tables
- Metadata
- hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be *accessible*?

- **extract characters correctly**
- interword spaces
- soft semantics
- alternative text
e.g., for Figures, Formulas, Tables
- Metadata
- hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ **extract characters correctly**
  US/English language generally OK — what about other languages and scripts?

▶ interword spaces

▶ soft semantics

▶ alternative text
  e.g., for Figures, Formulas, Tables

▶ Metadata

▶ hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  
  US/English language generally OK — what about other languages and scripts?
  
  Map to Unicode, wherever possible

- interword spaces

- soft semantics

- alternative text
  
  e.g., for Figures, Formulas, Tables

- Metadata

- hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  e.g., for Figures, Formulas, Tables

- **Metadata**

- **hard semantics**

  - PDF/A-
  - PDF/UA
  - `Matterhorn Protocol`
  - Acrobat Pro’s Accessibility tests
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ extract characters correctly
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

▶ interword spaces
  alastwomensink

▶ soft semantics

▶ alternative text
  e.g., for Figures, Formulas, Tables

▶ Metadata

▶ hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ *extract characters correctly*
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

▶ **interword spaces** alas two men sink

▶ soft semantics

▶ alternative text
e.g., for Figures, Formulas, Tables

▶ Metadata

▶ hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ extract characters correctly
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

▶ interword spaces	alas two men sink	a boating accident?

▶ soft semantics

▶ alternative text
e.g., for Figures, Formulas, Tables

▶ Metadata

▶ hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ extract characters correctly
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

▶ interword spaces alastwomensink

▶ soft semantics

▶ alternative text
  e.g., for Figures, Formulas, Tables

▶ Metadata

▶ hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- **interword spaces**
  - a last womens ink

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables

- **Metadata**

- **hard semantics**
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ **extract characters correctly**
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

▶ **interword spaces**  
  a last womens ink  
  stationery supplies?

▶ soft semantics

▶ alternative text
  e.g., for Figures, Formulas, Tables

▶ Metadata

▶ hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ extract characters correctly
   US/English language generally OK — what about other languages and scripts?
   Map to Unicode, wherever possible

▶ interword spaces

▶ soft semantics

▶ alternative text
   e.g., for Figures, Formulas, Tables

▶ Metadata

▶ hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**
  - Section headings, Lists, Tabular content

- **alternative text**
  - e.g., for Figures, Formulas, Tables

- **Metadata**

- **hard semantics**

- **PDF/A-#a**
- **PDF/UA**
- **Matterhorn Protocol**
- **Acrobat Pro’s Accessibility tests**
What is Accessibility?

What is involved in claiming an electronic document to be **accessible**?

- **extract characters correctly**
  
  US/English language generally OK — what about other languages and scripts?  
  
  Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  
  e.g., for Figures, Formulas, Tables

- **Metadata**

- **hard semantics**
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

► extract characters correctly
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

► interword spaces

► soft semantics

► alternative text
  e.g., for Figures, Formulas, Tables — not the same as a caption

► Metadata

► hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be *accessible*?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
    - Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**

- **hard semantics**
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  US/English language generally OK — what about other languages and scripts?  
  Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  e.g., Window title.

- **hard semantics**
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ extract characters correctly
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

▶ interword spaces

▶ soft semantics

▶ alternative text
  e.g., for Figures, Formulas, Tables — not the same as a caption

▶ Metadata
  e.g., Window title. Helps decide: is this the right file?

▶ hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  e.g., Window title. Helps decide: is this the right file? Do I want to read this?

- **hard semantics**
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ extract characters correctly
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

▶ interword spaces

▶ soft semantics

▶ alternative text
  e.g., for Figures, Formulas, Tables — not the same as a caption

▶ Metadata
  e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?

▶ hard semantics
What is Accessibility?

What is involved in claiming an electronic document to be *accessible*?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  - *Don’t underestimate the value of Metadata.*

- **hard semantics**
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  - Don’t underestimate the value of Metadata.

- **hard semantics**
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  - *Don’t underestimate the value of Metadata.*

- **hard semantics**
  - Footnotes
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  - Don’t underestimate the value of Metadata.

- **hard semantics**
  - Footnotes, References
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  - Don’t underestimate the value of Metadata.

- **hard semantics**
  - Footnotes, References, Hyperlinking
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- extract characters correctly
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- interword spaces

- soft semantics

- alternative text
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- Metadata
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  - Don’t underestimate the value of Metadata.

- hard semantics
  - Footnotes, References, Hyperlinking, Table cells
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  Don’t underestimate the value of Metadata.

- **hard semantics**
  Footnotes, References, Hyperlinking, Table cells, Table-of-Contents
What is Accessibility?

What is involved in claiming an electronic document to be *accessible*?

- **extract characters correctly**
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  Don’t underestimate the value of Metadata.

- **hard semantics**
  Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, …
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  - Don’t underestimate the value of Metadata.

- **hard semantics**
  - Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, …
  - PDF/A-#a
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ extract characters correctly
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

▶ interword spaces

▶ soft semantics

▶ alternative text
  e.g., for Figures, Formulas, Tables — not the same as a caption

▶ Metadata
  e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  Don’t underestimate the value of Metadata.

▶ hard semantics
  Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, …
  PDF/A-#a, PDF/UA
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

▶ extract characters correctly
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

▶ interword spaces

▶ soft semantics

▶ alternative text
  e.g., for Figures, Formulas, Tables — not the same as a caption

▶ Metadata
  e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  Don’t underestimate the value of Metadata.

▶ hard semantics
  Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, …
  PDF/A-#a, PDF/UA, ‘Matterhorn Protocol’
What is Accessibility?

What is involved in claiming an electronic document to be *accessible*?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?

  Don’t underestimate the value of Metadata.

- **hard semantics**
  - Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, …
  - PDF/A-#a, PDF/UA, ‘Matterhorn Protocol’, Acrobat Pro’s Accessibility tests
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible

- **interword spaces**

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information? Don’t underestimate the value of Metadata.

- **hard semantics**
  - Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, …
  - PDF/A-#a, PDF/UA, ‘Matterhorn Protocol’, Acrobat Pro’s Accessibility tests, …
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible

- interword spaces

- soft semantics

- alternative text
  e.g., for Figures, Formulas, Tables — not the same as a caption

- Metadata
  e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  Don’t underestimate the value of Metadata.

- hard semantics
  Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, …
  PDF/A-#a, PDF/UA, ‘Matterhorn Protocol’, Acrobat Pro’s Accessibility tests, …
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible — CMaps

- **interword spaces**

- **soft semantics**

- **alternative text**
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information?
  - Don’t underestimate the value of Metadata.

- **hard semantics**
  - Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, …
  - PDF/A-#a, PDF/UA, ‘Matterhorn Protocol’, Acrobat Pro’s Accessibility tests, …
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  US/English language generally OK — what about other languages and scripts?
  Map to Unicode, wherever possible  —  CMaps, Virtual fonts

- interword spaces

- soft semantics

- alternative text
  e.g., for Figures, Formulas, Tables  —  not the same as a caption

- Metadata
  e.g., Window title.  Helps decide: is this the right file? Do I want to read this? How to find related information?
  Don’t underestimate the value of Metadata.

- hard semantics
  Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, . . .
  PDF/A-#a, PDF/UA, Matterhorn Protocol’, Acrobat Pro’s Accessibility tests, . . .
What is Accessibility?

What is involved in claiming an electronic document to be accessible?

- **extract characters correctly**
  - US/English language generally OK — what about other languages and scripts?
  - Map to Unicode, wherever possible — CMaps, Virtual fonts, /ActualText

- interword spaces

- soft semantics

- alternative text
  - e.g., for Figures, Formulas, Tables — not the same as a caption

- **Metadata**
  - e.g., Window title. Helps decide: is this the right file? Do I want to read this? How to find related information? Don’t underestimate the value of Metadata.

- **hard semantics**
  - Footnotes, References, Hyperlinking, Table cells, Table-of-Contents, …
  - PDF/A-#a, PDF/UA, ‘Matterhorn Protocol’, Acrobat Pro’s Accessibility tests, …
NOW HERE’S SOME REAL-WORLD TEXT, USING ACCENTED CHARACTERS FROM EUROPEAN AND SLAVONIC LANGUAGES.

ICELANDIC: HÆÐIR SÉRHJÓÐAR TÁKNUÐU ANNARS TVÍHLJÓÐ

ENCYCLOPÆDIE: WOORDENBOEK VOOR WETENSCHAP EN KUNST, BESCHAVING EN NIJVERHEID. CHIEF EDITOR: A. WINKLER PRINS.
For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

- Why is that?
text-extraction

For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using T\TeX.

▶ Why is that? It depends upon the font used!
text-extraction

For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

- Why is that? It depends upon the font used!
- Can it be fixed?
For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

- Why is that? It depends upon the font used!
- Can it be fixed? Most \TeX fonts have problems of some kind.
For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

- Why is that? It depends upon the font used!
- Can it be fixed? Most \TeX fonts have problems of some kind.
- Yes, it can.
For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

- Why is that?  It depends upon the font used!
- Can it be fixed? Most \TeX fonts have problems of some kind.
- Yes, it can. mostly
text-extraction

For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

- Why is that? It depends upon the font used!
- Can it be fixed? Most \TeX fonts have problems of some kind.
- Yes, it can. mostly
- Is it easy to do?
text-extraction

For a very long time (always?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

- Why is that? It depends upon the font used!
- Can it be fixed? Most \TeX fonts have problems of some kind.
- Yes, it can. mostly
- Is it easy to do?
- Yes
For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

- Why is that? It depends upon the font used!
- Can it be fixed? Most \TeX fonts have problems of some kind.
- Yes, it can. mostly
- Is it easy to do?
- Yes (kind of)
text-extraction

For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX .

- Why is that? It depends upon the font used!
- Can it be fixed? Most \TeX fonts have problems of some kind.
- Yes, it can. mostly
- Is it easy to do?
- Yes (kind of), and No
For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

- Why is that? It depends upon the font used!
- Can it be fixed? Most \TeX{} fonts have problems of some kind.
- Yes, it can. mostly
- Is it easy to do?
- Yes (kind of), and No (for some things).
For a very long time (always?) there has been trouble with reliably extracting textual content from many PDFs produced using TeX.

- Why is that? It depends upon the font used!
- Can it be fixed? Most TeX fonts have problems of some kind.
- Yes, it can. mostly
- Is it easy to do?
- Yes (kind of), and No (for some things).
For a very long time (always?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

- **Why is that?** It depends upon the font used!
- **Can it be fixed?** Most \TeX fonts have problems of some kind.
- **Yes, it can.** mostly
- **Is it easy to do?**
- **Yes (kind of), and No (for some things).**
For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using \TeX.

\begin{itemize}
\item Why is that? It depends upon the font used!
\item Can it be fixed? Most \TeX\ fonts have problems of some kind.
\item Yes, it can. mostly
\item Is it easy to do?
\item Yes (kind of), and No (for some things).
\end{itemize}
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:
The font issues fall into 3 categories:

1. small capitals: need to be mapped to ordinary lowercase letters.

Lars Hellström, tex-fonts mailing list Nov. 2011
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. small capitals: need to be mapped to ordinary lowercase letters.

2. accents: need to come after the base they modify.
The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters.

2. **accents**: need to come after the base they modify.

3. **some individual characters**: need **special** treatment.

Lars Hellström, tex-fonts mailing list Nov. 2011
The font issues fall into 3 categories:

1. **small capitals** need to be mapped to ordinary lowercase letters.

2. accents: need to come after the base they modify.

3. some individual characters: need **special** treatment.
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals** need to be mapped to ordinary lowercase letters. This is done by attaching a CMap to a *font instance*.

2. accents: need to come after the base they modify.

3. some individual characters: need **special** treatment.

Lars Hellström, tex-fonts mailing list Nov. 2011
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals** need to be mapped to ordinary lowercase letters. This is done by attaching a CMap to a *font instance*; *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. accents: need to come after the base they modify.

3. some individual characters: need *special* treatment.

---

1 Lars Hellström, tex-fonts mailing list Nov. 2011
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters. This is done by attaching a CMap to a font instance; duplicating the (roman-style) (.tfm) font for small-caps virtual font.

2. **need to come after the base they modify**.

3. **some individual characters**: need special treatment. Here CMaps offer little help.

A little known feature of virtual fonts is that arbitrary PDF content can be included, using (SPECIAL ... ) instructions.  

Include (SPECIAL:direct:/SPAN <</ActualText (FEFF....)>> BDC) at the beginning of the character description, and (SPECIAL:direct:EMC) at the end. This latter technique doesn’t work with XETEX, since it doesn’t correctly implement \special{pdf:direct ...}.

Lars Hellström, tex-fonts mailing list Nov. 2011
The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters.
   This is done by attaching a CMap to a *font instance*; 
   *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents** need to come after the base they modify.

3. some individual characters: need **special** treatment.
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters.
   This is done by attaching a CMap to a *font instance*;
   *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents** need to come after the base they modify.
   Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap

3. some individual characters: need **special** treatment.
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters. This is done by attaching a CMap to a font instance; *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents** need to come after the base they modify. Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap; requiring the accent to be placed *after* the base.

3. some individual characters: need *special* treatment.

Lars Hellström, tex-fonts mailing list Nov. 2011
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters. This is done by attaching a CMap to a font instance; duplicating the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents** need to come after the base they modify. Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap; requiring the accent to be placed after the base, within the virtual font.

3. some individual characters: need special treatment.
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters. This is done by attaching a CMap to a font instance; *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents** need to come after the base they modify. Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap; requiring the accent to be placed *after* the base, within the virtual font. This applies to all T1-encoded (name ending -8t.vf) virtual fonts (and others too?).

3. some individual characters: need **special** treatment.

---

Lars Hellström, tex-fonts mailing list Nov. 2011
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters.  
   This is done by attaching a CMap to a font instance;  
   *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents**: need to come after the base they modify.  
   Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap;  
   requiring the accent to be placed *after* the base, within the virtual font.  
   This applies to all T1-encoded (name ending -8t.vf) virtual fonts (and others too ?).

3. some **individual characters** need special treatment.
The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters. This is done by attaching a CMap to a font instance; *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents**: need to come after the base they modify. Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap; requiring the accent to be placed *after* the base, within the virtual font. This applies to all T1-encoded (name ending -8t.vf) virtual fonts (and others too?).

3. some **individual characters** need special treatment. Here CMaps offer little help.
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters. This is done by attaching a CMap to a font instance; *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents**: need to come after the base they modify. Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap; requiring the accent to be placed *after* the base, within the virtual font. This applies to all T1-encoded (name ending -8t.vf) virtual fonts (and others too?).

3. some **individual characters** need special treatment. Here CMaps offer little help. A little known feature of virtual fonts is that arbitrary PDF content can included
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters.  
   This is done by attaching a CMap to a font instance;  
   *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents**: need to come after the base they modify.  
   Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap;  
   requiring the accent to be placed *after* the base, within the virtual font.  
   This applies to all T1-encoded (name ending -8t.vf) virtual fonts (and others too ?).

3. some **individual characters** need **special** treatment.  
   Here CMaps offer little help. A little known feature of virtual fonts is that arbitrary PDF content can included, using *(SPECIAL ... )* instructions.¹

¹ Lars Hellström, tex-fonts mailing list Nov. 2011
The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters.
   
   This is done by attaching a CMap to a *font instance*;
   *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents**: need to come after the base they modify.
   
   Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap;
   requiring the accent to be placed *after* the base, within the virtual font.
   This applies to all T1-encoded (name ending -8t.vf) virtual fonts (and others too ?).

3. some **individual characters** need **special** treatment.
   
   Here CMaps offer little help. A little known feature of virtual fonts is that arbitrary PDF content can included, using (SPECIAL ... ) instructions.¹
   
   include (SPECIAL:direct:/SPAN <</ActualText (FEFF....)>> BDC at the beginning of the character description

¹ Lars Hellström, tex-fonts mailing list Nov. 2011
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals:** need to be mapped to ordinary lowercase letters.
   This is done by attaching a CMap to a font instance;
   *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents:** need to come after the base they modify.
   Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap;
   requiring the accent to be placed *after* the base, within the virtual font.
   This applies to all T1-encoded (name ending -8t.vf) virtual fonts (and others too ?).

3. some **individual characters** need special treatment.
   Here CMaps offer little help. A little known feature of virtual fonts is that arbitrary PDF content can included, using *(SPECIAL ... )* instructions.¹
   include *(SPECIAL:direct:/SPAN <</ActualText (FEFF....)>> BDC)* at the beginning of the character description, and *(SPECIAL:direct:EMC)* at the end.

---

¹ Lars Hellström, tex-fonts mailing list Nov. 2011
CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

1. **small capitals**: need to be mapped to ordinary lowercase letters.
   This is done by attaching a CMap to a *font instance*;
   *duplicating* the (roman-style) (.tfm) font, within the small-caps virtual font.

2. **accents**: need to come after the base they modify.
   Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap;
   requiring the accent to be placed *after* the base, within the virtual font.
   This applies to all T1-encoded (name ending -8t.vf) virtual fonts (and others too ?).

3. **some individual characters**: need **special** treatment.
   Here CMaps offer little help. A little known feature of virtual fonts is that arbitrary PDF
   content can included, using (**SPECIAL ...**) instructions.¹
   include (**SPECIAL:direct:/SPAN <</ActualText (FEFF....)>> BDC**) at the beginning of the
   character description, and (**SPECIAL:direct:EMC**) at the end.

This latter technique doesn’t work with XEricTeX

¹ Lars Hellström, tex-fonts mailing list Nov. 2011
The font issues fall into 3 categories:

1. **small capitals:** need to be mapped to ordinary lowercase letters. This is done by attaching a CMap to a font instance; *duplicating* the (roman-style) (*.tfm*) font, within the small-caps virtual font.

2. **accents:** need to come after the base they modify. Accent characters need to be mapped to Unicode ‘combining accents’ within a CMap; requiring the accent to be placed *after* the base, within the virtual font. This applies to all T1-encoded (name ending -*8t.vf*) virtual fonts (and others too ?).

3. **some individual characters:** need *special* treatment. Here CMaps offer little help. A little known feature of virtual fonts is that arbitrary PDF content can included, using *(SPECIAL ... )* instructions.\(^1\) include *(SPECIAL:direct:/SPAN <</ActualText (FEFF....)>> BDC)* at the beginning of the character description, and *(SPECIAL:direct:EMC)* at the end.

This latter technique doesn’t work with *XeTEX*, since it doesn’t correctly implement \special{pdf:direct ...}.

---

\(^1\) Lars Hellström, tex-fonts mailing list Nov. 2011
CMaps

Adding a CMap to a font is relatively easy. PDFTEX can create CMaps automatically, using glyph names:

\pdfgentounicode=1

requires lists of the mappings of glyphs to unicode values: glyphtounicode.tex using

\pdfglyphtounicode{⟨glyph name⟩}{⟨Unicode string in Hex⟩}

dfx package provides extras: glyphtounicode-cmr.tex, glyphtounicode-ntx.tex.

PDFTEX can attach a CMap directly to a (tfm) font:

\def\attachCMap #1#2{% #1 = CMap file #2 = font TFM name
\immediate\pdfobj stream file{#1}\%
expan\pdffontattr #2{/ToUnicode \the\pdflastobj\space 0 R}\%
expan if\pdfnobuiltintounicode% LuaTEX doesn't have this primitive\else\pdfnobuiltintounicode #2\relax\fi\%
\pdfincludechars #2{a \char"20}\%
}\font\bchsmallcaps=bchr8rs scaled 800
\attachCMap{bchsc.cmap}{\bchsmallcaps}
\pdfmapline{= bchr8rs CharterBT-Roman " TeXBase1Encoding ReEncodeFont " <8r.enc <bchr8a.pfb}

XETEX adds the CMap le directly to the (PDF) font:

\special{pdf:mapline bchr8r 8r.enc bchr8a.pfb -u bchr8r.cmap}\
\special{pdf:mapline bchr8rs 8r.enc bchr8a.pfb -u bchsc.cmap}:%
CMaps

Adding a CMap to a font is relatively easy.
Adding a CMap to a font is relatively easy.

- pdfTeX can create CMaps automatically, using glyph names:
  \pdfgentounicode=1
  \begin{verbatim}
  \let\pdfglyphtounicode\expandafter\pdfglyphtounicode
  \end{verbatim}
  requires lists of the mappings of glyphs to unicode values:
  \begin{verbatim}
  glyphptounicode.tex
  \end{verbatim}

- pdfx package provides extras:
  \begin{verbatim}
  glyphptounicode-cmr.tex, glyphptounicode-ntx.tex.
  \end{verbatim}
Adding a CMap to a font is relatively easy.

- **pdfTeX** can create CMAPS automatically, using glyph names:
  
  ```latex
  \pdfgentounicode=1
  ``

  requires lists of the mappings of glyphs to unicode values:
  ```latex
  \pdfglyphtounicode{{\langle glyph name} \rangle}{\langle Unicode string in Hex} \rangle
  ``

  **pdfx** package provides extras: `glyphtounicode-cmr.tex`, `glyphtounicode-ntx.tex`.

- **pdfTeX** can attach a CMap directly to a (.tfm) font:
  ```latex
  \def\attachCMap #1#2{% #1 = CMap file #2 = font TFM name
  \immediate\pdfobj stream file{#1}\
  \expandafter\pdffontattr #2{/ToUnicode \the\pdflastobj\space 0 R}\
  \expandafter\ifx\csname pdfnobuiltintounicode\endcsname\relax
  % LuaTeX doesn’t have this primitive
  \else
  \expandafter\pdffontattr #2{/ToUnicode \the\pdflastobj\space 0 R}\
  \fi
  \pdfincludechars #2{a \char"20}% ensure the font is not discarded
  }
  \font\bchsmallcaps=bchr8rs scaled 800
  \attachCMap{bchsc.cmap}{\bchsmallcaps}
  \pdfmapline{= bchr8rs CharterBT-Roman " TeXBase1Encoding ReEncodeFont " <8r.enc <bchr8a.pfb}
Adding a CMap to a font is relatively easy.

- pdfTeX can create CMaps automatically, using glyph names:
  \pdfgentounicode=1
  \texttt{glyphtounicode.tex}

  requires lists of the mappings of glyphs to unicode values:
  \texttt{glyphtounicode{⟨glyph name⟩}{⟨Unicode string in Hex⟩}}

- pdfx package provides extras: \texttt{glyphtounicode-cmr.tex}, \texttt{glyphtounicode-ntx.tex}.

- pdfTeX can attach a CMap directly to a (.tfm) font:
  \begin{verbatim}
  \def\attachCMap #1#2{%
    \immediate\pdfobj stream file{#1}\
    \expandafter\pdffontattr #2{/ToUnicode \the\pdflastobj\space 0 R}%
    \expandafter\ifx\csname pdfnobuiltintounicode\endcsname\relax
      % LuaTeX doesn't have this primitive
    \else
      \expandafter\pdfnobuiltintounicode #2\relax
    \fi
    \pdfincludechars #2{a \char"20}% ensure the font is not discarded
  }
  \font\bchsmallcaps=bchr8rs scaled 800
  \attachCMap{bchsc.cmap}{\bchsmallcaps}
  \pdfmapline{= bchr8rs CharterBT-Roman " TeXBase1Encoding ReEncodeFont " <8r.enc <bchr8a.pfb}
  \end{verbatim}

- \LaTeX{} adds the CMap file directly to the (PDF) font:
  \begin{verbatim}
  \special{pdf:mapline bchr8r 8r.enc bchr8a.pfb -u bchr8r.cmap}%
  \special{pdf:mapline bchr8rs 8r.enc bchr8a.pfb -u bchsc.cmap}%
  \end{verbatim}
CMap for fake small-caps fonts

The difficult part is knowing which is the .tfm font, to which the CMap must be attached. For example, bchr8c.vf is the virtual font for small-caps in Charter font. But attachCMap{bchsc.cmap}{bchr8c} does not work!!

The human-readable form bchr8c.vpl of the virtual font bchr8c.vf starts as:

\begin{verbatim}
(MAPFONT D 0
 (FONTNAME bchr8r)
 (FONTCHECKSUM O 1764027361)
 (FONTAT R 0.8)
 (FONTDSIZE R 10.0)
)
(MAPFONT D 1
 (FONTNAME bchr8r)
 (FONTCHECKSUM O 1764027361)
 (FONTAT R 1.0)
 (FONTDSIZE R 10.0)
)
\end{verbatim}

No CMap can make this correct. Both upper and lowercase letters are drawn from the same PDF font, so all glyph names will be uppercased ones; just shown at different sizes.

Change it to:

\begin{verbatim}
(MAPFONT D 0
 (FONTNAME bchr8rs)
 (FONTCHECKSUM O ... ...
 (FONTAT R 0.8)
 (FONTDSIZE R 10.0)
)
(MAPFONT D 1
 (FONTNAME bchr8r)
 (FONTCHECKSUM O 1764027361)
 (FONTAT R 1.0)
 (FONTDSIZE R 10.0)
)
\end{verbatim}

attach bchsc.cmap to bchr8rs, with CMap sending all uppercase letters into lowercase ones (i.e., reversing the lower \rightarrow upper of small capitals)

Now attachCMap{bchsc.cmap}{bchr8rs} works, having two separate font instances.
CMap for fake small-caps fonts

The difficult part is knowing which is the .tfm font, to which the CMap must be attached. For example, `bchr8c.vf` is the virtual font for small-caps in Charter font. But `\attachCMap{bchsc.cmap}{bchr8c}` does not work!!

Change it to:

```
\attach{bchsc.cmap}{bchr8rs}
```

Now `\attachCMap{bchsc.cmap}{bchr8rs}` works, having two separate font instances.

---

2 Volovich, V.; cmap package README
CMap for fake small-caps fonts

The difficult part is knowing which is the \texttt{.tfm} font, to which the CMap must be attached. For example, \texttt{bchr8c.vf} is the virtual font for small-caps in Charter font. But \texttt{\attachCMap\{bchsc.cmap\}\{bchr8c\}} does not work!! ²

The human-readable form \texttt{bchr8c.vpl} of the virtual font \texttt{bchr8c.vf} starts as:

\begin{verbatim}
(MAPFONT D 0
 (FONTNAME bchr8r)
 (FONTCHECKSUM O 1764027361)
 (FONTAT R 0.8)
 (FONTDSIZE R 10.0)
 )

(MAPFONT D 1
 (FONTNAME bchr8r)
 (FONTCHECKSUM O 1764027361)
 (FONTAT R 1.0)
 (FONTDSIZE R 10.0)
 )
\end{verbatim}

No CMap can make this correct. Both upper and lowercase letters are drawn from the same PDF font, so all glyph names will be uppercased ones; just shown at different sizes.

² Volovich, V.; cmap package README
CMap for fake small-caps fonts

The difficult part is knowing which is the .tfm font, to which the CMap must be attached. For example, bchr8c.vf is the virtual font for small-caps in Charter font. But \attachCMap{bchsc.cmap}{bchr8c} does not work!!

The human-readable form bchr8c.vpl of the virtual font bchr8c.vf starts as:

```
(MAPFONT D 0
 (FONTNAME bchr8r)
 (FONTCHECKSUM O 1764027361)
 (FONTAT R 0.8)
 (FONTDSIZE R 10.0)
 )
(MAPFONT D 1
 (FONTNAME bchr8r)
 (FONTCHECKSUM O 1764027361)
 (FONTAT R 1.0)
 (FONTDSIZE R 10.0)
 )
```

No CMap can make this correct. Both upper and lowercase letters are drawn from the same PDF font, so all glyph names will be uppercased ones; just shown at different sizes.

Change it to:

```
(MAPFONT D 0
 (FONTNAME bchr8rs)
 (FONTCHECKSUM O ... ...)
 (FONTAT R 0.8)
 (FONTDSIZE R 10.0)
 )
(MAPFONT D 1
 (FONTNAME bchr8r)
 (FONTCHECKSUM O 1764027361)
 (FONTAT R 1.0)
 (FONTDSIZE R 10.0)
 )
```

attach bchsc.cmap to bchr8rs, with CMap sending all uppercase letters into lowercase ones (i.e., reversing the lower \(\rightarrow\) upper of small capitals)

\(^2\)Volovich, V.; cmap package README
CMap for fake small-caps fonts

- The difficult part is knowing which is the .tfm font, to which the CMap must be attached. For example, bchr8c.vf is the virtual font for small-caps in Charter font. But \attachCMap{bchsc.cmap}{bchr8c} does not work!!

- The human-readable form bchr8c.vpl of the virtual font bchr8c.vf starts as:

  ```
  (MAPFONT D 0
   (FONTNAME bchr8r)
   (FONTCHECKSUM O 1764027361)
   (FONTAT R 0.8)
   (FONTDSIZE R 10.0)
  )
  (MAPFONT D 1
   (FONTNAME bchr8r)
   (FONTCHECKSUM O 1764027361)
   (FONTAT R 1.0)
   (FONTDSIZE R 10.0)
  )
  ```

  No CMap can make this correct. Both upper and lowercase letters are drawn from the same PDF font, so all glyph names will be uppercased ones; just shown at different sizes.

- Change it to:

  ```
  (MAPFONT D 0
   (FONTNAME bchr8rs)
   (FONTCHECKSUM O ... ...)
   (FONTAT R 0.8)
   (FONTDSIZE R 10.0)
  )
  (MAPFONT D 1
   (FONTNAME bchr8r)
   (FONTCHECKSUM O 1764027361)
   (FONTAT R 1.0)
   (FONTDSIZE R 10.0)
  )
  ```

  attach bchsc.cmap to bchr8rs, with CMap sending all uppercase letters into lowercase ones (i.e., reversing the lower → upper of small capitals)

- Now \attachCMap{bchsc.cmap}{bchr8rs} works, having two separate font instances.

---

2 Volovich, V.; cmap package README
CMap for fake small-caps fonts

The difficult part is knowing which is the .tfm font, to which the CMap must be attached. For example, bchr8c.vf is the virtual font for small-caps in Charter font. But 

\[ \text{attachCMap}{bchsc.cmap}{bchr8c} \]

does not work!!

The virtual font bchr8c.vf starts as:

\[ \begin{align*}
&'\text{ONT D 1} \\
&'\text{ONTNAME bchr8r} \\
&'\text{ONTCHECKSUM 0 1764027361} \\
&'\text{ONTAT R 1.0} \\
&'\text{ONTDSIZE R 10.0} \\
\end{align*} \]

The upper and lowercase letters are drawn from the uppercased ones; just shown at different sizes.

\[ \begin{align*}
&'\text{ONT D 1} \\
&'\text{ONTNAME bchr8r} \\
&'\text{ONTCHECKSUM 0 1764027361} \\
&'\text{ONTAT R 1.0} \\
&'\text{ONTDSIZE R 10.0} \\
\end{align*} \]

\textit{sending all uppercase letters into lowercase small capitals} \)

\[ \text{\text{attachCMap}{bchsc.cmap}{bchr8rs}} \]

works, having two separate font instances.

---

2 Volovich, V.; cmap package README
accents coming after the base

- Accented characters, such as Ä typically are described in a virtual font as at left:
Accented characters, such as Ņ typically are described in a virtual font as at left:

\[
\begin{align*}
&M (\text{MAP}) \\
&M \hspace{1cm} (\text{MOVERIGHT R 0.025}) \\
&M \hspace{1cm} (\text{PUSH}) \\
&M \hspace{1cm} (\text{MOVEDOWN R -0.198}) \\
&M \hspace{1cm} (\text{MOVERIGHT R 0.07}) \\
&M \hspace{1cm} (\text{SELECTFONT D 1}) \\
&M \hspace{1cm} (\text{SETCHAR O 13}) \\
&M \hspace{1cm} (\text{POP}) \\
&M \hspace{1cm} (\text{SETCHAR C A}) \\
&M \hspace{1cm} (\text{MOVERIGHT R 0.025}) \]

When this is changed to at right above, there is no visual difference within the PDF, but now the breve accent character (in slot 013 = 112224) is placed after the base A, so will be extracted after it.

Use a CMap that associates the breve accent with the Unicode 'combining acute accent' at U+0306. Note that this CMap must be associated with the font that references. Frequently this can be allowed to be generated automatically, if its glyph names are standard.

In a T1-encoded font, there can be as many as 105 instances of an accented-over letter, where the accent should be moved, in this way, to coming after the base.
accents coming after the base

▶ Accented characters, such as Ũ typically are described in a virtual font as at left:

(MAP
  (MOVERIGHT R 0.025)
  (PUSH)
  (MOVEDOWN R -0.198)
  (MOVERIGHT R 0.07)
  (SELECTFONT D 1)
  (SETCHAR O 13)
  (POP)
  (SETCHAR C A)
  (MOVERIGHT R 0.025)
)

▶ When this is changed to at right above, there is no visual difference within the PDF, but now the breve accent character (in slot \013 = 11) is placed after the base ‘A’, so will be extracted after it.
accents coming after the base

▶ Accented characters, such as Ā typically are described in a virtual font as at left:

```plaintext
(MAP
  (MOVERIGHT R 0.025)
  (PUSH)
  (MOVEDOWN R -0.198)
  (MOVERIGHT R 0.07)
  (SELECTFONT D 1)
  (SETCHAR O 13)
  (POP)
  (MOVERIGHT R 0.025)
  (MOVERIGHT R 0.07)
)  (MAP
  (PUSH)
  (MOVERIGHT R 0.025)
  (SELECTFONT D 1)
  (SETCHAR C A)
  (MOVERIGHT R 0.025)
  (MOVERIGHT R 0.07)
  (SETCHAR O 13)
  (POP)
  (MOVEDOWN R -0.198)
)```

▶ When this is changed to at right above, there is no visual difference within the PDF, but now the breve accent character (in slot \013 = 11) is placed after the base ‘A’, so will be extracted after it.

▶ Use a CMap that associates the breve accent with the Unicode ‘combining acute accent’ at U+0306. Note that this CMap must be associated with the font that (SELECTFONT D 1) references. Frequently this can be allowed to be generated automatically, if its glyph names are standard.
Accented characters, such as Ā typically are described in a virtual font as at left:

```
(MAP
  (MOVERIGHT R 0.025)
  (PUSH)
  (MOVEDOWN R -0.198)
  (MOVERIGHT R 0.07)
  (SELECTFONT D 1)
  (SETCHAR C A)
  (MOVERIGHT R 0.025)
)
```

When this is changed to at right above, there is no visual difference within the PDF, but now the breve accent character (in slot \013 = 11) is placed after the base ‘A’, so will be extracted after it.

Use a CMap that associates the breve accent with the Unicode ‘combining acute accent’ at U+0306. Note that this CMap must be associated with the font that (SELECTFONT D 1) references. Frequently this can be allowed to be generated automatically, if its glyph names are standard.

In a T1-encoded font, there can be as many as 105 instances of an accented-over letter, where the accent should be moved, in this way, to coming after the base.
/ActualText within a virtual font

Lars Hellström, tex-fonts mailing list Nov. 2011
For this final technique, I’m indebted to Lars Hellström’s posting\(^3\), even though the example coding does not actually achieve the ‘visible space’ character that was desired.

\(^3\)Lars Hellström, tex-fonts mailing list Nov. 2011
For this final technique, I’m indebted to Lars Hellström’s posting\(^3\), even though the example coding does not actually achieve the ‘visible space’ character that was desired.

\(^3\)Lars Hellström, tex-fonts mailing list Nov. 2011
For this final technique, I’m indebted to Lars Hellström’s posting\textsuperscript{3}, even though the example coding does not actually achieve the ‘visible space’ character that was desired.

Lars’ code (at left) does not work upon extracting text because there is no actual font character to be selected, so the `/ActualText` replacement string will never be used.

\textsuperscript{3}Lars Hellström, tex-fonts mailing list Nov. 2011
For this final technique, I’m indebted to Lars Hellström’s posting\(^3\), even though the example coding does not actually achieve the ‘visible space’ character that was desired.

Lars’ code (at left) does not work upon extracting text because there is no actual font character to be selected, so the `/ActualText` replacement string will never be used.

In the coding at right, we use pdfTeX’s ‘fake space’ font via `(SELECTFONT D 2). It is this character that gets selected, and mapped to the ‘visible space’ at U+2423.

\(^3\) Lars Hellström, tex-fonts mailing list Nov. 2011
For this final technique, I’m indebted to Lars Hellström’s posting\textsuperscript{3}, even though the example coding does not actually achieve the ‘visible space’ character that was desired.

\begin{quote}
\texttt{(CHARACTER D 32 (COMMENT visiblespace) (CHARWD R 0.6) (CHARDP R 0.200) (MAP (SPECIAL pdf:direct:
/Span</ActualText<FEFF2423>>>BDC) (PUSH) (SETCHAR D 32) (COMMENT space) (POP) (MOVEUP R -0.2) (MOVERIGHT R 0.05) (SETRULE R 0.2 R 0.05) (SETRULE R 0.05 R 0.4) (SETRULE R 0.2 R 0.05) (MOVERIGHT R 0.05) (MOVEDOWN R 0.2) (SPECIAL pdf:direct:EMC) )}
\end{quote}

\begin{quote}
\texttt{(CHARACTER O 40 (COMMENT visible space) (CHARWD R 0.67198) (CHARDP R 0.2085) (MAP (SPECIAL pdf:direct:
/Span</ActualText<FEFF2423>>>BDC) (SELECTFONT D 2) (SETCHAR O 40) (MOVERIGHT R 0.025) (MOVEDOWN R 0.2) (MOVERIGHT R 0.05) (SETRULE R 0.2 R 0.061) (SETRULE R 0.061 R 0.4) (SETRULE R 0.2 R 0.061) (MOVERIGHT R 0.05) (MOVEDOWN R -0.2) (MOVERIGHT R 0.025) )}
\end{quote}

\textsuperscript{3}Lars Hellström, tex-fonts mailing list Nov. 2011

Lars’ code (at left) does not work upon extracting text because there is no actual font character to be selected, so the \texttt{/ActualText} replacement string will never be used. In the coding at right, we use pdfTEX’s `fake space’ font via \texttt{(SELECTFONT D 2)}. It is this character that gets selected, and mapped to the ‘visible space’ at U+2423.
Examples, IJ, dotless-i, and more:

(CHARACTER 0 234 (COMMENT IJ )
 (CHARWD R 0.86798)
 (CHARHT R 0.665)
 (CHARDP R 0.014)
 (COMMENT
   (KRN C A R -0.037)
   ...
 )
 (MAP
   (SPECIAL pdf:direct:
     /Span<</ActualText<FEFF0132>>>BDC)
   (MOVERIGHT R 0.025)
   (SELECTFONT D 1)
   (SETRULE R 0.482 R 0.0)
   (SPECIAL pdf:direct:EMC)
 )
)

(CHARACTER 0 27 (COMMENT ZWNJ )
 (CHARWD R 0.0)
 (CHARHT R 0.491)
 (MAP
   (SPECIAL pdf:direct:
     /Span<</ActualText<FEFF200C>>>BDC)
   (SELECTFONT D 2)
   (SETRULE R 0.482 R 0.0)
   (SPECIAL pdf:direct:EMC)
 )
)

(CHARACTER 0 31 (COMMENT dotlessi )
 (CHARWD R 0.30899)
 (CHARHT R 0.5545)
 (MAP
   (SPECIAL pdf:direct:
     /Span<</ActualText<FEFF0131>>>BDC)
   (MOVERIGHT R 0.05)
   (SETRULE R 0.482 R 0.0)
   (SPECIAL pdf:direct:EMC)
 )
)
Other characters can be given suitable mappings to Unicode using this technique.

Examples, IJ, dotless-i, and more:

```latex
(CHARACTER 0 234 (COMMENT IJ )
  (CHARWD R 0.86798)
  (CHARHT R 0.665)
  (CHARDP R 0.014)
  (COMMENT
    (KRN C A R -0.037)
    ...
  )
  (MAP
    (SPECIAL pdf:direct:
      /Span<</ActualText<FEFF0132>>>BDC)
    (SELECTFONT D 1)
    (SETCHAR C I)
    (MOVERIGHT R 0.025)
  )
)
```

```latex
(CHARACTER 0 27 (COMMENT ZWNJ )
  (CHARWD R 0.0)
  (CHARHT R 0.491)
  (MAP
    (SPECIAL pdf:direct:
      /Span<</ActualText<FEFF200C>>>BDC)
    (SELECTFONT D 2)
    (SETCHAR 0 40)
    (SETRULE R 0.482 R 0.0)
    (SPECIAL pdf:direct:EMC)
  )
)
```

```latex
(CHARACTER 0 31 (COMMENT dotlessi )
  (CHARWD R 0.30899)
  (CHARHT R 0.5545)
  (MAP
    (SPECIAL pdf:direct:
      /Span<</ActualText<FEFF0131>>>BDC)
    (MOVERIGHT R 0.025)
    (MOVERIGHT R 0.05)
    (MOVERIGHT R 0.025)
    (SPECIAL pdf:direct:EMC)
  )
)
```

This technique allows you to override a mapping to Unicode, from a CMap or based upon glyph name, should it be appropriate to do so.
Other characters can be given suitable mappings to Unicode using this technique.

Examples, IJ, dotless-i, and more:

```
(CHARACTER 0 234 (COMMENT IJ )
  (CHARWD R 0.86798)
  (CHARHT R 0.665)
  (CHARDP R 0.014)
  (COMMENT
    (KRN C A R -0.037)
    ...
  )
  (MAP
    (SPECIAL pdf:direct:
      /Span<</ActualText<FEFF200C>>>BDC)
    (SELECTFONT D 2)
    (SETCHAR 0 40)
    (SETRULE R 0.482 R 0.0)
    (SPECIAL pdf:direct:EMC)
  )
)
```

```
(CHARACTER 0 27 (COMMENT ZWNJ )
  (CHARWD R 0.0)
  (CHARHT R 0.491)
  (MAP
    (SPECIAL pdf:direct:
      /Span<</ActualText<FEFF200C>>>BDC)
    (SELECTFONT D 2)
    (SETCHAR 0 40)
    (SETRULE R 0.482 R 0.0)
    (SPECIAL pdf:direct:EMC)
  )
)
```

```
(CHARACTER 0 31 (COMMENT dotlessi )
  (CHARWD R 0.30899)
  (CHARHT R 0.5545)
  (MAP
    (SPECIAL pdf:direct:
      /Span<</ActualText<FEFF200C>>>BDC)
    (MOVERIGHT R 0.025)
    (SELECTFONT D 1)
    (SETCHAR C I)
    (MOVERIGHT R 0.05)
    (SETCHAR C J)
    (MOVERIGHT R 0.025)
    (SPECIAL pdf:direct:EMC)
  )
)
```

This technique allows you to override a mapping to Unicode, from a CMap or based upon glyph name, should it be appropriate to do so.

Drawback: it only works with pdfTeX, and maybe also LuaTeX.