

Accessibility-related talks

Ross Moore, Macquarie University, Sydney
TUG 2020, Online: 25–27 July 2020





JULY 24-26

2020

TEX AND L^AT_EX
TYPOGRAPHY
TYPESETTING
FONTS
DESIGN
PUBLISHING
AND MORE

41ST ANNUAL CONFERENCE OF THE T_EX USERS GROUP

Accessibility, text-extraction

Accessibility, author advice for Tagged PDF

Tagging in TikZ diagrams.

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

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

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**

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, ...

text-extraction

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?

text-extraction

For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using $\text{T}_{\text{E}}\text{X}$.

- ▶ 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 $\text{T}_{\text{E}}\text{X}$.

- ▶ 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 $\text{T}_{\text{E}}\text{X}$.

- ▶ Why is that? It depends upon the font used!
- ▶ Can it be fixed? Most $\text{T}_{\text{E}}\text{X}$ 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 $\text{T}_{\text{E}}\text{X}$.

- ▶ Why is that? It depends upon the font used!
- ▶ Can it be fixed? Most $\text{T}_{\text{E}}\text{X}$ fonts have problems of some kind.
- ▶ Yes, it can.

text-extraction

For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using $\text{T}_{\text{E}}\text{X}$.

- ▶ Why is that? It depends upon the font used!
- ▶ Can it be fixed? Most $\text{T}_{\text{E}}\text{X}$ fonts have problems of some kind.
- ▶ Yes, it can. mostly

For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using $\text{T}_{\text{E}}\text{X}$.

- ▶ Why is that? It depends upon the font used!
- ▶ Can it be fixed? Most $\text{T}_{\text{E}}\text{X}$ fonts have problems of some kind.
- ▶ Yes, it can. mostly
- ▶ Is it easy to do?

For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using $\text{T}_{\text{E}}\text{X}$.

- ▶ Why is that? It depends upon the font used!
- ▶ Can it be fixed? Most $\text{T}_{\text{E}}\text{X}$ 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 $\text{T}_{\text{E}}\text{X}$.

- ▶ Why is that? It depends upon the font used!
- ▶ Can it be fixed? Most $\text{T}_{\text{E}}\text{X}$ fonts have problems of some kind.
- ▶ Yes, it can. mostly
- ▶ Is it easy to do?
- ▶ Yes (kind of)

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 $\text{T}_{\text{E}}\text{X}$.

- ▶ Why is that? It depends upon the font used!
- ▶ Can it be fixed? Most $\text{T}_{\text{E}}\text{X}$ 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 $\text{T}_{\text{E}}\text{X}$.

- ▶ Why is that? It depends upon the font used!
- ▶ Can it be fixed? Most $\text{T}_{\text{E}}\text{X}$ fonts have problems of some kind.
- ▶ Yes, it can. mostly
- ▶ Is it easy to do?
- ▶ Yes (kind of), and No (for some things).

KOU 169 METRU SE JEDNA O NEJKRATŠI PRAŽSKÝ MOST.
HIČ HIR KELIME Ğ ILE BAŞLAMAZ! HIÇ HIR ZAMAN İLK SIRADA
AĞAC KELEMEŞİN OYATASINDA, DİĞ.

SŁOWAK: NAPOSLEDY HEJANÉ SŁOVÁ:
Ľ, DEPRIMÁCIA, PREPER, PREPER, WICHĆIĆ, WICHŁASTANÉ, DEPRIVÁ
DEPRIVOVÁ, DEPRIVOVANÝ, NAŠKUBANÍ, KOWÁŁKA, LEŻĄCI, SMÍČEN
NEOKRÓCHANÝ, NALOŽIT, OKRÓCHANÝ, OKRÓCHANEC, PRĘGAZDOW
LÁRÁWÁ, ZRETELNI, SUPLIK, ŚUPLIKA, CHOR, RASŁAWICE, POTRÍH
POLIETRE, WINDASNÁŻÉ SA, HODOWÁRIAN, ZWREŁZOVÁ, ZMOČN
NŹ, FRAGERBŔCKIN, ZDZEŔAĆ, PLAGIE, ODĆAROWANI, CEPER, NALÉ
MIZMUS, VEČER, PRACHÁR, KLAVÍR, PEPŔITI, ĞARŁOWI, FORMAN, SI
SORIT, ĞLOHA, MŁSAŔ, ZAZIWĆOWÁ, XTETO, SKRÓTTETEĽNI, POZ
MŁSA, MŁSKA, ZAWIEST, ZAWIET, MRŔŔAK, ZAWIEST, PAMŁSKO, OST
SBA, TULÁĆT SA, EKSPIRÁCIA, ESPIRÁCIA, CIWŔWÉ, HINĆOWÓW, I
ŔEKO, ŰTKOŠT, WESEŁA, DOLNA KRUKA, DOSŁOŽANÉ, WELJENKTI
MEZITI, FOSTREĆKOWÁWÁ, TAKTIEZ, PAPUKISKO, SŔNŔW SA, USER
EVA, ASIMILACJA, ZDICHÁWÁ, MORAWANĆIN, HOTEL, WIDÉŔŔI, EK
LAPINDA, WEZEŔŔI, UĆUTI, NATESAĆ, ĆLOWESTWI, CHŁESODARCA, P

POLISH: POŁOŻONE JEST W CENTRUM POLSKI, NA SKRZYŻOWANI
DOWYCH I KOLEJOWYCH CIĄGÓW KOMUNIKACYJNYCH. GRANICZY
INNYMI WOJEWÓDZTWAMI: KAZDOWICKIM, ŚWIĘTOKRZYSKIM, ŚL
SKIM, WIELKOPOLSKIM I KJAWIŔSKO-POMORSKIM.

ADMINISTRACYJNIE WJEWÓDZTWO JEST PODZIELONE NA 177 G
3 MIASTA NA PRAWACH POWIATU) I 21 POWIATÓW. TRZY MIASTA

‘the bad’

(p.1, p.4)

(p.1, p.2, p.9)

text-extraction

For a very long time (always ?) there has been trouble with reliably extracting textual content from many PDFs produced using $\text{T}_{\text{E}}\text{X}$.

- ▶ Why is that? It depends upon the font used!
- ▶ Can it be fixed? Most $\text{T}_{\text{E}}\text{X}$ fonts have problems of some kind.
- ▶ Yes, it can. mostly
- ▶ Is it easy to do?
- ▶ Yes (kind of), and No (for some things).

KOU 169 METRU SE JEDNA O NEJLEPŠÍ PRAŽSKÝ MOST.

HIČ HIR KELIME Ğ İLE BAŞLAMAZ! HIÇ HIR ZAMAN İLE SIRADA AĞAC KEMELERİN OYUNUNA, DAĞ.

SŁOWA: NAFOSLIZY WEADANE SŁOWA:

Ł, DEPRIMACJA, PRZEPR, PRZEPR, WIECZCIĆ, WYCHŁASTANI, DEPRWA DEPRIVOWAN, DEPRIVOWAN, NAKĘBIANI, KOWÁŁKA, LEZACI, SMIEĆEN OKRÓCHYANI, NALOZIT, OKRÓCHYANI, OKRÓCHANE, PRZEĞADOW LAKARAT, ZMETELNI, SUPLIK, ŠUPLIKA, CHOR, RASŁAWICE, POTRBI POLETRZE, WYNASNAJEZ SA, HODOROBRIANI, ZYBRELZOWAT, ZMOČEN NIĆ, FRAGEREČIN, ZDEKAT, PLAGIAR, ODCAROWANI, CEFER, NALÉ MIZMUS, VEČER, PRACNAR, KLAVIR, PRFPIZIT, ŠARLOWI, FORMAN, SI SORIT, GŁOHA, MŁSZA, ZAZIWCOWAT, İKTETO, SKRÓTTEDELNI, POZ MŁSA, MŁSKA, ZAWIEST, ZAWIEZ, MRNAN, ZAWIEST, PAMLSOK, OSTI SKA, TULACZT SA, EKSPIRACJA, ESPIRACJA, CIHWANÉ, HINČOWAN, I NERO, UTKOSY, WESKLA, DOŁNA KRUPA, DOSZADZANI, VELEJENKTY MEZITI, POSTREČKOWANAT, TAKTICE, PAPSUKID, SNIWAT SA, USER EVA, ASSIMILACJA, ZDICHAWAT, MORAWANČINI, HOTEL, WIDEDERŃI, EK LAPINDA, WEZERN, UČITI, NATESAT, ČLOWESTWI, CHLEBODARCA, P

POLSK: POŁOŻONE JEST W CENTRUM POLSKI, NA SKRZYŻOWANI DOWYCH I KOLEJOWYCH CIĄGÓW EDMINIKACYJNYCH. GRANICZY INNYMI WOLEWÓDZTWAMI: RĄDOWICKIM, ŚWIĘTOBRZYWSKIM, ŚL ŚKIM, WIELKOPOLSKIM I KUJAWSKO-POMORSKIM.

ADMINISTRACYJNIE WOLEWÓDZTWO JEST PODZIELONE NA 177 G 3 MIASTA NA PRACACH POWIATU I 21 POWIATÓW. TRZY MIASTA

	0	1	2	3	4	5	6	
00x	.	'	'	'	'	'	'	'
01x	'	'	'	'	'	'	'	'
02x	'	'	'	'	'	'	'	'
03x	■	■	■	■	■	■	■	■
04x	■	■	■	■	■	■	■	■
05x	()	'	'	'	'	'	'
06x	0	1	2	3	4	5	6	
07x	8	9	'	'	'	'	'	'
08x	@	A	B	C	D	E	F	G
09x	H	I	J	K	L	M	N	O
10x	P	Q	R	S	T	U	V	W
11x	X	Y	Z	[\]	^	_
12x	·	·	·	·	·	·	·	·
13x	·	·	·	·	·	·	·	·
14x	·	·	·	·	·	·	·	·
15x	·	·	·	·	·	·	·	·
16x	·	·	·	·	·	·	·	·
17x	·	·	·	·	·	·	·	·
18x	·	·	·	·	·	·	·	·
19x	·	·	·	·	·	·	·	·
20x	·	·	·	·	·	·	·	·
21x	·	·	·	·	·	·	·	·
22x	·	·	·	·	·	·	·	·
23x	·	·	·	·	·	·	·	·
24x	·	·	·	·	·	·	·	·
25x	·	·	·	·	·	·	·	·
26x	·	·	·	·	·	·	·	·
27x	·	·	·	·	·	·	·	·
28x	·	·	·	·	·	·	·	·
29x	·	·	·	·	·	·	·	·
30x	·	·	·	·	·	·	·	·

'the bad'

'the good' (p.1, p.4)

(p.1, p.2, p.9)

text-extraction

For a very long time (always ?) there has been trouble with reliably extracting textual content. from many PDFs produced using $\text{T}_\text{E}_\text{X}$.

- ▶ Why is that? It depends upon the font used!
- ▶ Can it be fixed? Most $\text{T}_\text{E}_\text{X}$ fonts have problems of some kind.
- ▶ Yes, it can. mostly
- ▶ Is it easy to do?
- ▶ Yes (kind of), and No (for some things).

KOU 169 METRU SE JEDNA O NEJLEPŠÍCH PRAŽSKÝCH MOSTŮ.

HIC BIR KELIME Ğ İLE BAĞLAMAZI HİÇ BİR ZAMAN İLE SIRADA AĞAC KEMERİNİN OTURASINDA, DAĞ.

SLOVAK: NÁPOJILYV HĎADANÉ SLOVÁ:

Ľ, DEPRIMÁČKA, PREPER, PREPER, WICHČÍŤ, WICHĽASTANÍ, DEPRVÁ DEPRIVOVÁŤ, DEPRIVOVANÝ, NÁŠRUBANÍ, KOVÁĽKA, LEŽÁČ, SMIEČI NÁOKRÓCHYANÝ, NALOŽÍŤ, OKRÓCHYANÝ, OKRÓCHANEČ, PREĎAZOV LÁKÁVAT, ŽRETELNÝ, SUPĽIK, ŠŮPLIKA, CHOR, RASĽAVICE, POTRŤI POLEŤTE, WINDASNÁŽEŤ SA, HODODRŤAN, ZVERĽEĽOVÁŤ, ŽMOČNŤ, FRAGERČSKIN, ZDEKÁŤ, PLAGIAR, OĎČORANÝ, ČEPER, NÁĽ MIZMUS, VEČER, PRACHÁR, KLAVIR, PEPIŤI, ĎARĽOVÍ, FORMAN, SI SOBÍŤ, ĽOHA, MĽSÁŤ, ZAZVICOVÁŤ, EXTETO, SKRÓTTEDĽNÍ, POZ MĽSÁ, MĽSKÁ, ZAVIEŠŤ, ZAVIEŽŤ, MŤNÁK, ZAVIEŠŤ, PAMĽOSK, OSTI SBA, TULÁČEŤ SA, EKSPIRÁČKA, ESPIRÁČKA, ČIHWÁŤ, HŤCŤOVANÝ, I NĚKO, ŤKŤOSŤ, VESEĽÁ, DOĽNÁ KRUPA, DOSAZĽANÍ, VEĽKĽENŤI MIZITI, POSTREČKŤOVÁŤ, TAKTIEZ, PÁPUŠKIDŤ, SŤNÁV SA, UER EVA, ASSIMILÁČKA, ZDICHÁVÁŤ, MORAVANČIN, HOTEL, WĽDEĽNÍ, KEX LAPINDA, WEŽEŤNÍ, ŤCŤÍŤI, NATESÁŤ, ČĽOWEŠŤWÍ, CHĽOGODARCA, P

POLISH: POŁOŻONE JEST W CENTRUM POLSKI, NA SKRZYŻOWARCI DROWY I KOLEJOWYCH CIĄGŤW KDMUNIKACYJNYCH. GRANICY DNINY WOLEWŤÓZTWANE: RĄDOWICKIM, ŚWIEŤOZYRSKIM, ŚĽSKIM, WIELKOPOLSKIM I KJUMWKO-POMORSKIM.

ADMINISTRACYJNIE WOLEWŤÓZTWO JEST PODZIELONE NA 177 G 3 MIASTA NA PRAWACH POWIATU I 21 POWIATŤW. TRZY MIASTA

	0	1	2	3	4	5	6
D0x	+	-	+	-	+	-	+
D1x	-	+	-	+	-	+	-
D2x	+	-	+	-	+	-	+
D3x	■	□	△	▽	◇	○	×
D4x	⌘	⌘	⌘	⌘	⌘	⌘	⌘
D5x	()	[]	{	}	~
D6x	O	1	2	3	4	5	6
D7x	S	9	1	2	<	>	>
D8x	@	A	B	C	D	E	F
D9x	H	I	J	K	L	M	N
Dzx	P	Q	R	S	T	U	V
Dxx	X	Y	Z	[\]	^
D4x	+	-	+	-	+	-	+
D5x	H	I	J	K	L	M	N
D6x	P	Q	R	S	T	U	V
D7x	X	Y	Z	[\]	^
D8x	+	-	+	-	+	-	+
D9x	H	I	J	K	L	M	N
Dzx	P	Q	R	S	T	U	V
Dxx	X	Y	Z	[\]	^
D4x	+	-	+	-	+	-	+

Copy bchr8r.tfm renamed as bchr8ra, and build a new virtual table that differs from bchr8r only in using this bchr8ra for the lower

(COPYRIGHT © 1994) (COPYRIGHT © 1994)
(FONTFORGE.COM © 1764027361) (FONTFORGE.COM © 1764027361)
(FONTS & G.U.) (FONTS & G.U.)
(FONTSIZE & 10.0) (FONTSIZE & 10.0)

	0	1	2	3	4	5	6
D0x	+	-	+	-	+	-	+
D1x	-	+	-	+	-	+	-
D2x	+	-	+	-	+	-	+
D3x	■	□	△	▽	◇	○	×
D4x	⌘	⌘	⌘	⌘	⌘	⌘	⌘
D5x	()	[]	{	}	~
D6x	O	1	2	3	4	5	6
D7x	S	9	1	2	<	>	>
D8x	@	A	B	C	D	E	F
D9x	H	I	J	K	L	M	N
Dzx	P	Q	R	S	T	U	V
Dxx	X	Y	Z	[\]	^
D4x	+	-	+	-	+	-	+

'the bad'

'the good' (p.1, p.4)

How it is done. (p.1, p.2, p.9)

CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

CMaps, Virtual Fonts, /ActualText fixes

The font issues fall into 3 categories:

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

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.

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.
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.
2. accents: need to come after the base they modify.
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*
2. accents: need to come after the base they modify.
3. some individual characters: need **special** treatment.

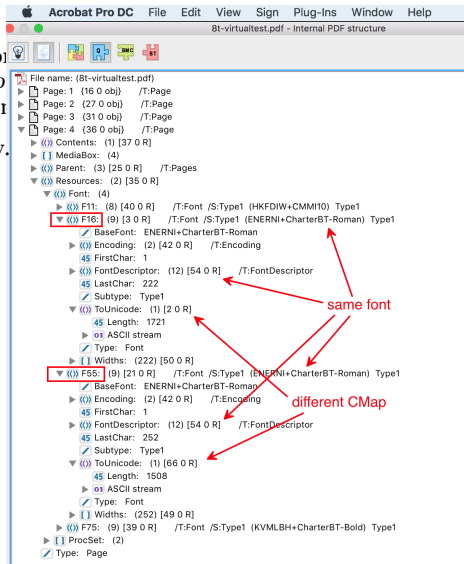
The font issues fall into 3 categories:

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.
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 old style. This is done by attaching a CMap to a font, *for duplicating* the (roman-style) (.tfm) font.
2. need to come after the base they modify.
3. some individual characters:



Acrobat Pro DC File Edit View Sign Plug-Ins Window Help

8t-virtualtest.pdf - Internal PDF structure

File name: (8t-virtualtest.pdf)

- Page: 1 (16 0 obj) /T:Page
- Page: 2 (27 0 obj) /T:Page
- Page: 3 (31 0 obj) /T:Page
- Page: 4 (36 0 obj) /T:Page
 - Contents: (1) [37 0 R]
 - MediaBox: (4)
 - Parent: (3) [25 0 R] /T:Pages
 - Resources: (2) [35 0 R]
 - Font: (4)
 - F11: (8) [40 0 R] /T:Font /S:Type1 (HKFDIW+CMM10) Type1
 - F16: (9) [3 0 R] /T:Font /S:Type1 (ENERNI+CharterBT-Roman) Type1**
 - BaseFont: ENERNI+CharterBT-Roman
 - Encoding: (2) [42 0 R] /T:Encoding
 - FirstChar: 1
 - FontDescriptor: (12) [54 0 R] /T:FontDescriptor
 - LastChar: 222
 - Subtype: Type1
 - ToUnicode: (1) [2 0 R]
 - Length: 1721
 - ASCII stream
 - Type: Font
 - Widths: (222) [50 0 R]
 - F55: (9) [21 0 R] /T:Font /S:Type1 (ENERNI+CharterBT-Roman) Type1**
 - BaseFont: ENERNI+CharterBT-Roman
 - Encoding: (2) [42 0 R] /T:Encoding
 - FirstChar: 1
 - FontDescriptor: (12) [54 0 R] /T:FontDescriptor
 - LastChar: 252
 - Subtype: Type1
 - ToUnicode: (1) [66 0 R]
 - Length: 1508
 - ASCII stream
 - Type: Font
 - Widths: (252) [49 0 R]
 - F75: (9) [39 0 R] /T:Font /S:Type1 (KVMLBH+CharterBT-Bold) Type1
 - ProcSet: (2)
 - Type: Page

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.

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.

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.

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.

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) (.t₁fm) 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.

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

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 be included, using (`SPECIAL ...`) instructions.¹

¹Lars 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

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 be 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

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 be 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 X₃TeX

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 be 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 X_YTeX, since it doesn’t correctly implement `\special{pdf:direct ...}`.

¹ Lars Hellström, tex-fonts mailing list Nov. 2011

CMaps

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:
requires lists of the mappings of glyphs to unicode values:
using `\pdfglyptounicode{<glyph name>}{<Unicode string in Hex>}`
pdfx package provides extras: `glyptounicode-cmr.tex`, `glyptounicode-ntx.tex`.
`\pdfgentounicode=1`
`glyptounicode.tex`

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>}`
pdfx 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}%
\expandafter\pdffontattr #2{/ToUnicode \the\pdflastobj\space 0 R}%
\expandafter\ifx\csname pdfnobluiltintounicode\endcsname\relax
% LuaTeX doesn't have this primitive
\else
\expandafter\pdfnobluiltintounicode #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}
```


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>}`
pdfx 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}%
\expandafter\pdffontattr #2{/ToUnicode \the\pdflastobj\space 0 R}%
\expandafter\ifx\csname pdfnobluiltintounicode\endcsname\relax
% LuaTeX doesn't have this primitive
\else
\expandafter\pdfnobluiltintounicode #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}
```

- ▶ XeTeX adds the CMap file 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}%
```

CMap for fake small-caps fonts

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!!²

²Volovich, 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!!²
- ▶ The human-readable form `bchr8c.vpl` of the virtual font `bchr8c.vf` starts as:

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

²Volovich, 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!!²
- ▶ The human-readable form `bchr8c.vpl` of the virtual font `bchr8c.vf` starts as:

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

²Volovich, 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!!²
- ▶ The human-readable form `bchr8c.vpl` of the virtual font `bchr8c.vf` starts as:

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

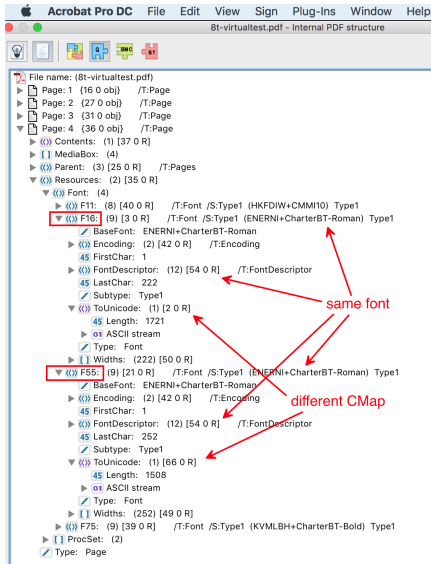
```
(MAPFONT D 0                                (MAPFONT D 1
  (FONTNAME bchr8rs)                          (FONTNAME bchr8r)
  (FONTCHECKSUM 0 ... ...)                   (FONTCHECKSUM 0 1764027361)
  (FONTAT R 0.8)                             (FONTAT R 1.0)
  (FONTDSIZE R 10.0)                      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

²Volovich, V.; cmap package [README](#)

CMap for fake small-caps fonts



File name: (8t-virtualtest.pdf)

- Page: 1 (16 0 obj) /T:Page
- Page: 2 (27 0 obj) /T:Page
- Page: 3 (31 0 obj) /T:Page
- Page: 4 (36 0 obj) /T:Page
- Contents: (1) [37 0 R]
- MediaBox: (4)
- Parent: (3) [25 0 R] /T:Pages
- Resources: (2) [35 0 R]
 - Font: (4)
 - F11: (8) [40 0 R] /T:Font /S:Type1 (HKFDIW+CMM10) Type1
 - F16: (9) [3 0 R] /T:Font /S:Type1 (ENERNI+CharterBT-Roman) Type1**
 - BaseFont: ENERNI+CharterBT-Roman
 - Encoding: (2) [42 0 R] /T:Encoding
 - 45 FirstChar: 1
 - FontDescriptor: (12) [54 0 R] /T:FontDescriptor
 - 45 LastChar: 222
 - Subtype: Type1
 - ToUnicode: (1) [2 0 R]
 - 45 Length: 1721
 - ASCII stream
 - Type: Font
 - Widths: (222) [50 0 R]
 - F55: (9) [21 0 R] /T:Font /S:Type1 (ENERNI+CharterBT-Roman) Type1**
 - BaseFont: ENERNI+CharterBT-Roman
 - Encoding: (2) [42 0 R] /T:Encoding
 - 45 FirstChar: 1
 - FontDescriptor: (12) [54 0 R] /T:FontDescriptor
 - 45 LastChar: 252
 - Subtype: Type1
 - ToUnicode: (1) [66 0 R]
 - 45 Length: 1508
 - ASCII stream
 - Type: Font
 - Widths: (252) [49 0 R]
 - F75: (9) [39 0 R] /T:Font /S:Type1 (KVMLBH+CharterBT-Bold) Type1
 - ProcSet: (2)
 - Type: Page

tfm font, to which the CMap must be attached.

or small-caps in Charter font.

does not work!!²

ie virtual font bchr8c.vf starts as:

```

ONT D 1
ONTNAME bchr8r)
ONTCHECKSUM 0 1764027361)
ONTAT R 1.0)
ONTDSIZE R 10.0)

```

r and lowercase letters are drawn from the uppercase ones; just shown at different sizes.

```

ONT D 1
ONTNAME bchr8r)
ONTCHECKSUM 0 1764027361)
ONTAT R 1.0)
ONTDSIZE R 10.0)

```

ending all *uppercase* letters into lowercase small capitals)

works, having two separate font *instances*.

²Volovich, V.; cmap package [README](#)

accents coming after the base

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

accents coming after the base

- ▶ Accented characters, such as Å 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)
)

(MAP

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

accents coming after the base

- ▶ Accented characters, such as Å 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)
)
```

(MAP

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

- ▶ 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 Å 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)
)
```

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

- ▶ 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.

accents coming after the base

- ▶ Accented characters, such as Å 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 0 13)
(POP)
(SETCHAR C A)
(MOVERIGHT R 0.025)
)
```

```
(MAP
(PUSH)
(MOVERIGHT R 0.025)
(SELECTFONT D 1)
(SETCHAR C A)
(MOVERIGHT R 0.025)
(POP)
(MOVEDOWN R -0.198)
(MOVERIGHT R 0.07)
(SETCHAR 0 13)
)
```

- ▶ When this is changed to at right above, there is *no visual difference* within the PDF, but now the breve accent character (in slot $\backslash 013 = 11$) 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

/ActualText within a virtual font

- ▶ For this final technique, I'm indebted to Lars Hellström's posting³, even though the example coding does not actually achieve the 'visible space' character that was desired.

³Lars Hellström, tex-fonts mailing list [Nov. 2011](#)

/ActualText within a virtual font

- ▶ For this final technique, I'm indebted to Lars Hellström's posting³, even though the example coding does not actually achieve the 'visible space' character that was desired.

```
▶ (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)
    (MOVEUP R 0.2)
    (SPECIAL pdf:direct:EMC)
  )
)

(CHARACTER 0 40 (COMMENT visible space )
  (CHARWD R 0.67198)
  (CHARDP R 0.2085)
  (MAP
    (SPECIAL pdf:direct:
      /Span<</ActualText<FEFF2423>>>BDC)
    (SELECTFONT D 2)
    (SETCHAR 0 40)
    (MOVERIGHT R 0.025)
    (MOVEDOWN R 0.2)
    (MOVERIGHT R 0.05)
    (SPECIAL pdf:direct:EMC)
    (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)
  )
)
```

³Lars Hellström, tex-fonts mailing list Nov. 2011

/ActualText within a virtual font

- ▶ For this final technique, I'm indebted to Lars Hellström's posting³, even though the example coding does not actually achieve the 'visible space' character that was desired.

- ▶

<pre>(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) (MOVEUP R 0.2) (SPECIAL pdf:direct:EMC)))</pre>	<pre>(CHARACTER 0 40 (COMMENT visible space) (CHARWD R 0.67198) (CHARDP R 0.2085) (MAP (SPECIAL pdf:direct: /Span<</ActualText<FEFF2423>>>BDC) (SELECTFONT D 2) (SETCHAR 0 40) (MOVERIGHT R 0.025) (MOVEDOWN R 0.2) (MOVERIGHT R 0.05) (SPECIAL pdf:direct:EMC) (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)))</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- ▶ 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.

³Lars Hellström, tex-fonts mailing list Nov. 2011

/ActualText within a virtual font

- ▶ For this final technique, I'm indebted to Lars Hellström's posting³, even though the example coding does not actually achieve the 'visible space' character that was desired.

<pre> ▶ (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) (MOVEUP R 0.2) (SPECIAL pdf:direct:EMC))) </pre>	<pre> (CHARACTER 0 40 (COMMENT visible space) (CHARWD R 0.67198) (CHARDP R 0.2085) (MAP (SPECIAL pdf:direct: /Span<</ActualText<FEFF2423>>>BDC) (SELECTFONT D 2) (SETCHAR 0 40) (MOVERIGHT R 0.025) (MOVEDOWN R 0.2) (MOVERIGHT R 0.05) (SPECIAL pdf:direct:EMC) (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))) </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- ▶ 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.

³Lars Hellström, tex-fonts mailing list [Nov. 2011](#)

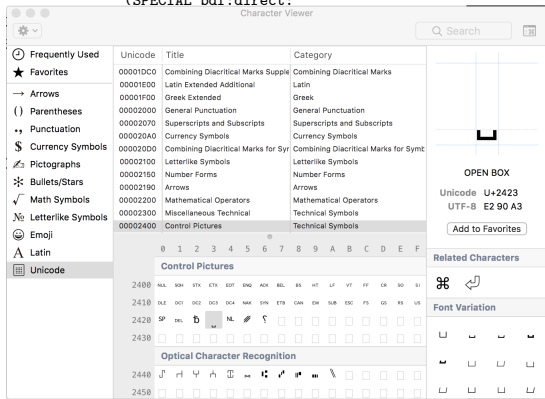
/ActualText within a virtual font

- ▶ For this final technique, I'm indebted to Lars Hellström's posting³, even though the example coding does not actually achieve the 'visible space' character that was desired.

- ▶ (CHARACTER D 32 (COMMENT visiblespace)
(CHARWD R 0.6)
(CHARDP R 0.200)
(MAP
(SPECIAL pdf:direct:

```
(CHARACTER 0 40 (COMMENT visible space )
(CharWD R 0.67198)
(CharDP R 0.2085)
(MAP
```

```
(SPECIAL pdf:direct:
```



Character Viewer

Search

Unicode	Title	Category
00001D0C	Combining Diacritical Marks Suppl	Combining Diacritical Marks
00001E00	Latin Extended Additional	Latin
00001F00	Greek Extended	Greek
00002000	General Punctuation	General Punctuation
00002070	Superscripts and Subscripts	Superscripts and Subscripts
000020A0	Currency Symbols	Currency Symbols
000020D0	Combining Diacritical Marks for Sym	Combining Diacritical Marks for Symt
00002100	Letterlike Symbols	Letterlike Symbols
00002150	Number Forms	Number Forms
00002190	Arrows	Arrows
00002200	Mathematical Operators	Mathematical Operators
00002300	Miscellaneous Technical	Technical Symbols
00002400	Control Pictures	Technical Symbols

OPEN BOX
Unicode U+2423
UTF-8 E2 90 A3
Add to Favorites

Related Characters

Font Variation

```
/Span<</ActualText<FEFF2423>>>BDC)
```

```
SELECTFONT D 2)
SETCHAR 0 40)
(MOVERIGHT R 0.025)
(MOVEDOWN R 0.2)
(MOVERIGHT R 0.05)
(SPECIAL pdf:direct:EMC)
(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)
```

... because there is no actual font
... cement string will never be used.

... font via (SELECTFONT D 2).
... the 'visible space' at U+2423 .

³ Lars Hellström, tex-fonts mailing list Nov. 2011

/ActualText for ligatures, etc.

/ActualText for ligatures, etc.

-
- ▶ Other characters can be given suitable mappings to Unicode using this technique.

/ActualText for ligatures, etc.

- ▶ 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<FEFF0132>>>>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)
  )
)
```

```
(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 dotless i )
  (CHARWD R 0.30899)
  (CHARHT R 0.5545)
  (MAP
    (SPECIAL pdf:direct:
      /Span<</ActualText<FEFF0131>>>>BDC)
    (MOVERIGHT R 0.025)
    (SETCHAR C I)
    (MOVERIGHT R 0.025)
    (SPECIAL pdf:direct:EMC)
  )
)
```

/ActualText for ligatures, etc.

- ▶ 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<FEFF0132>>>>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)
  )
)
```

```
(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 dotless i )
  (CHARWD R 0.30899)
  (CHARHT R 0.5545)
  (MAP
    (SPECIAL pdf:direct:
      /Span<</ActualText<FEFF0131>>>>BDC)
    (MOVERIGHT R 0.025)
    (SETCHAR C I)
    (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.

/ActualText for ligatures, etc.

- ▶ 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<FEFF0132>>>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)
)
```

```
(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 dotless i )
  (CHARWD R 0.30899)
  (CHARHT R 0.5545)
  (MAP
    (SPECIAL pdf:direct:
      /Span<</ActualText<FEFF0131>>>BDC)
    (MOVERIGHT R 0.025)
    (SETCHAR C I)
    (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 pdf_{TEX}, and maybe also Lua_{TEX}.

Accessibility, text-extraction

Accessibility, author advice for Tagged PDF

Tagging in TikZ diagrams.

Accessibility support with pdfT_EX

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

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

▶ **extract characters correctly**

Map to Unicode, wherever possible

▶ **interword spaces**

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

▶ **alternative text**

e.g., for Figures, Formulas, Tables

▶ **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**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps

handled by pdfx package

▶ **interword spaces**

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

▶ **alternative text**

e.g., for Figures, Formulas, Tables

▶ **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**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros

handled by pdfx package
handled by pdfx package

▶ **interword spaces**

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

▶ **alternative text**

e.g., for Figures, Formulas, Tables

▶ **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**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

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

▶ **alternative text**

e.g., for Figures, Formulas, Tables

▶ **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**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

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

▶ **alternative text**

e.g., for Figures, Formulas, Tables

▶ **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**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

▶ **alternative text**

e.g., for Figures, Formulas, Tables

▶ **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**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

handled by tpdf package

▶ **alternative text**

e.g., for Figures, Formulas, Tables

▶ **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**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

handled by tpdf package

▶ **alternative text**

e.g., for Figures, Formulas, Tables

handled by tpdf package

▶ **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**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

handled by tpdf package

▶ **alternative text**

e.g., for Figures, Formulas, Tables

handled by tpdf package

▶ **Metadata**

e.g., Window title.

handled by pdfx package

Helps decide: is this the right file? Do I want to read this? How to find related information?

▶ **hard semantics**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

handled by tpdf package

▶ **alternative text**

e.g., for Figures, Formulas, Tables

handled by tpdf package

▶ **Metadata**

e.g., Window title.

handled by pdfx package

Helps decide: is this the right file? Do I want to read this? How to find related information?

Use XMP Metadata

▶ **hard semantics**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

handled by tpdf package

▶ **alternative text**

e.g., for Figures, Formulas, Tables

handled by tpdf package

▶ **Metadata**

e.g., Window title.

Helps decide: is this the right file? Do I want to read this? How to find related information?

handled by pdfx package

Use XMP Metadata

handled by pdfx package

▶ **hard semantics**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

handled by tpdf package

▶ **alternative text**

e.g., for Figures, Formulas, Tables

handled by tpdf package

▶ **Metadata**

e.g., Window title.

handled by pdfx package

Helps decide: is this the right file? Do I want to read this? How to find related information?

Use XMP Metadata

handled by pdfx package

▶ **hard semantics**

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

handled by tpdf package

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

handled by tpdf package

▶ **alternative text**

e.g., for Figures, Formulas, Tables

handled by tpdf package

▶ **Metadata**

e.g., Window title.

Helps decide: is this the right file? Do I want to read this? How to find related information?

handled by pdfx package

Use XMP Metadata

handled by pdfx package

▶ **hard semantics**

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

handled by tpdf package

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

handled by tpdf package

▶ **alternative text**

e.g., for Figures, Formulas, Tables

handled by tpdf package

▶ **Metadata**

e.g., Window title.

Helps decide: is this the right file? Do I want to read this? How to find related information?

handled by pdfx package

Use XMP Metadata

handled by pdfx package

▶ **hard semantics**

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

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

handled by tpdf package

handled by tpdf package

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

handled by tpdf package

▶ **alternative text**

e.g., for Figures, Formulas, Tables

handled by tpdf package

▶ **Metadata**

e.g., Window title.

handled by pdfx package

Helps decide: is this the right file? Do I want to read this? How to find related information?

Use XMP Metadata

handled by pdfx package

▶ **hard semantics**

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

handled by tpdf package

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

handled by tpdf package

Accessibility support with pdfT_EX

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

▶ **extract characters correctly**

Map to Unicode, wherever possible auto-generated CMaps
accents after base with macros
other characters

handled by pdfx package

handled by pdfx package

more work to be done — see my previous talk

▶ **interword spaces**

`\pdfinterwordspaceon`, `\pdfinterwordspaceoff`

handled by tpdf package

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

handled by tpdf package

▶ **alternative text**

e.g., for Figures, Formulas, Tables

handled by tpdf package

▶ **Metadata**

e.g., Window title.

handled by pdfx package

Helps decide: is this the right file? Do I want to read this? How to find related information?

Use XMP Metadata

handled by pdfx package

▶ **hard semantics** — some of it can be *author-definable*

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

handled by tpdf package

PDF/A-#a, PDF/UA, 'Matterhorn Protocol', Acrobat Pro's Accessibility tests, ...

handled by tpdf package

Author advice, preparatory to tagging

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*, know the 'ins-and-outs' of the document format, such as tagging for 'Accessibility'.

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*, know the 'ins-and-outs' of the document format, such as tagging for 'Accessibility'.
- ▶ That's what editors are for! ... and some authors *will* want to!

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*, know the 'ins-and-outs' of the document format, such as tagging for 'Accessibility'.
- ▶ That's what editors are for! ... and some authors *will* want to!
- ▶ Nevertheless, there are some things that document authors *can do better*

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*, know the 'ins-and-outs' of the document format, such as tagging for 'Accessibility'.
- ▶ That's what editors are for! ... and some authors *will* want to!
- ▶ Nevertheless, there are some things that document authors *can do better*, to help streamline the extra editing and processing that is required to meet new standards, such as PDF/A and PDF/UA.

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*, know the 'ins-and-outs' of the document format, such as tagging for 'Accessibility'.
- ▶ That's what editors are for! ... and some authors *will* want to!
- ▶ Nevertheless, there are some things that document authors *can do better*, to help streamline the extra editing and processing that is required to meet new standards, such as PDF/A and PDF/UA.
- ▶ For example:

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*, know the 'ins-and-outs' of the document format, such as tagging for 'Accessibility'.
- ▶ That's what editors are for! ... and some authors *will* want to!
- ▶ Nevertheless, there are some things that document authors *can do better*, to help streamline the extra editing and processing that is required to meet new standards, such as PDF/A and PDF/UA.
- ▶ For example:
 - provide **alternative text** for Figures and Tables

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*, know the 'ins-and-outs' of the document format, such as tagging for 'Accessibility'.
- ▶ That's what editors are for! ... and some authors *will* want to!
- ▶ Nevertheless, there are some things that document authors *can do better*, to help streamline the extra editing and processing that is required to meet new standards, such as PDF/A and PDF/UA.
- ▶ For example:
 - provide **alternative text** for Figures and Tables
 - hide physical markup in **user-defined macros**

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*, know the 'ins-and-outs' of the document format, such as tagging for 'Accessibility'.
- ▶ That's what editors are for! ... and some authors *will* want to!
- ▶ Nevertheless, there are some things that document authors *can do better*, to help streamline the extra editing and processing that is required to meet new standards, such as PDF/A and PDF/UA.
- ▶ For example:
 - provide **alternative text** for Figures and Tables
 - hide physical markup in **user-defined macros**
 - use **consistent naming** for such user-defined macros

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*, know the 'ins-and-outs' of the document format, such as tagging for 'Accessibility'.
- ▶ That's what editors are for! ... and some authors *will* want to!
- ▶ Nevertheless, there are some things that document authors *can do better*, to help streamline the extra editing and processing that is required to meet new standards, such as PDF/A and PDF/UA.
- ▶ For example:
 - provide **alternative text** for Figures and Tables
 - hide physical markup in **user-defined macros**
 - use **consistent naming** for such user-defined macros
 - identify significant semantic elements, using **environments and/or macros**

Author advice, preparatory to tagging

- ▶ Tagging a PDF document is *not* something that every author should be expected to do.
- ▶ For example, scientists want to do their own work, experimenting and collecting data, and write it down using familiar methods (perhaps using \LaTeX).
- ▶ A scientist will do a large part of the preparation of writing up their work, to ensure fidelity and accuracy of what is presented, but ...
- ▶ ... they don't want to, *nor should need to*, know the 'ins-and-outs' of the document format, such as tagging for 'Accessibility'.
- ▶ That's what editors are for! ... and some authors *will* want to!
- ▶ Nevertheless, there are some things that document authors *can do better*, to help streamline the extra editing and processing that is required to meet new standards, such as PDF/A and PDF/UA.
- ▶ For example:
 - provide **alternative text** for Figures and Tables
 - hide physical markup in **user-defined macros**
 - use **consistent naming** for such user-defined macros
 - identify significant semantic elements, using **environments and/or macros**
- ▶ Let's have a look at how to do this kind of thing, with a real-world example.

Documentation & example document

Documentation & example document

The document at left is the one submitted as a preprint for this talk.

Tagging with \LaTeX — Part 1: author considerations

Ross Moore*

Abstract

Successful tagging within PDF files generated from \LaTeX source encourages a change in viewpoint on the nature and intent of the \LaTeX coding. Using explicit examples from a real-world document, we illustrate how to capture such a change within the \LaTeX source, for various structural elements. Other issues for creating archival and accessible PDF documents are discussed.

1 Introduction

With “Tagged PDF” being the accepted method for creating PDF documents enriched to satisfy Accessibility requirements [3, 6], this article is intended to address the main issues that authors and editors should be aware of, with regards to tagging and \LaTeX usage. Examples are taken from a real-world fully-tagged research report, prepared in \LaTeX but also employing extra coding written by the author, in a package named `tpdf` that handles the technical aspects of producing “Tagged PDF”. That research report is the one used by the author in the talk [7] at TUG 2019, and delivered remotely using Zoom

be managed to ensure long-term preservation; ...

By producing documents conforming to published standards both PDF/A [3] and PDF/UA [4], these obligations can be met in full, if not surpassed.

2 Tagging commands for special content

It is a common typographical practice to use different styling to present names of books, magazines, or publications which have a particular relevance to the topic under discussion. Certainly the fact of a different style being used indicates to a fully-sighted reader that there is a special significance, but not what that significance actually is. That has to be deduced from context. With tagging, that significance can be made explicit. And with \LaTeX source this is very easy to do.

For example, the Night Skies Project report has a page which mentions the various Acts of Congress which underpin their work; see Figure 1. The names of these acts appear in *Italics*. This was originally done by simply specifying

```
\textit{Organic Act of 1916} ...
```

Changing this by inventing a macro `\srpaact` the true intention is captured, at least within the \LaTeX source. For typesetting purposes the expansion of

Advice for authors of \LaTeX documents needing to be Tagged.

Documentation & example document

The document at left is the one submitted as a preprint for this talk. It contains a description of tagging features that can be obtained, with advice to authors on how to make the tagging process easier to achieve.

Tagging with L^AT_EX — Part 1: author considerations

Ross Moore*

Abstract

Successful tagging within PDF files generated from L^AT_EX source encourages a change in viewpoint on the nature and intent of the L^AT_EX coding. Using explicit examples from a real-world document, we illustrate how to capture such a change within the L^AT_EX source, for various structural elements. Other issues for creating archival and accessible PDF documents are discussed.

1 Introduction

With “Tagged PDF” being the accepted method for creating PDF documents enriched to satisfy Accessibility requirements [3, 6], this article is intended to address the main issues that authors and editors should be aware of, with regards to tagging and L^AT_EX usage. Examples are taken from a real-world fully-tagged research report, prepared in L^AT_EX but also employing extra coding written by the author, in a package named `tpdf` that handles the technical aspects of producing “Tagged PDF”. That research report is the one used by the author in the talk [7] at TUG 2019, and delivered remotely using Zoom

be managed to ensure long-term preservation; ...

By producing documents conforming to published standards both PDF/A [3] and PDF/UA [4], these obligations can be met in full, if not surpassed.

2 Tagging commands for special content

It is a common typographical practice to use different styling to present names of books, magazines, or publications which have a particular relevance to the topic under discussion. Certainly the fact of a different style being used indicates to a fully-sighted reader that there is a special significance, but not what that significance actually is. That has to be deduced from context. With tagging, that significance can be made explicit. And with L^AT_EX source this is very easy to do.

For example, the Night Skies Project report has a page which mentions the various Acts of Congress which underpin their work; see Figure 1. The names of these acts appear in *Italics*. This was originally done by simply specifying

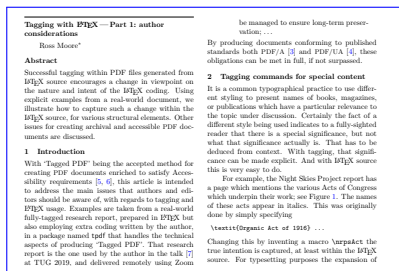
```
\textit{Organic Act of 1916} ...
```

Changing this by inventing a macro `\srpaact` the true intention is captured, at least within the L^AT_EX source. For typesetting purposes the expansion of

Advice for authors of L^AT_EX documents needing to be Tagged.

Documentation & example document

The document at left is the one submitted as a preprint for this talk. It contains a description of tagging features that can be obtained, with advice to authors on how to make the tagging process easier to achieve.



Tagging with L^AT_EX — Part 1: author considerations
Ross Moore*

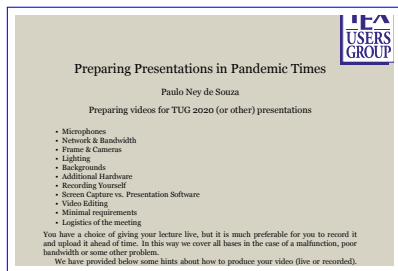
Abstract
Successful tagging within PDF files generated from L^AT_EX source encourages a change in viewpoint on the nature and intent of the L^AT_EX coding. Using explicit examples from a real-world document, we illustrate how to capture such a change within the L^AT_EX source, for various structural elements. Other issues for creating archival and accessible PDF documents are discussed.

1 Introduction
With “Tagged PDF” being the accepted method for creating PDF documents enriched to satisfy Accessibility requirements [3, 6], this article is intended to address the main issues that authors and editors should be aware of, with regards to tagging and L^AT_EX usage. Examples are taken from a real-world fully-tagged research report, prepared in L^AT_EX but also employing extra coding written by the author, in a package named `tpdf` that handles the technical aspects of producing “Tagged PDF”. That research report is the one used by the author in the talk [7] at TUG 2019, and delivered remotely using Zoom

be managed to ensure long-term preservation; ...
By producing documents conforming to published standards both PDF/A [3] and PDF/UA [4], these obligations can be met in full, if not surpassed.

2 Tagging commands for special content
It is a common typographical practice to use different styling to present names of books, magazines, or publications which have a particular relevance to the topic under discussion. Certainly the fact of a different style being used indicates to a fully-sighted reader that there is a special significance, but not what that significance actually is. That has to be deduced from context. With tagging, that significance can be made explicit. And with L^AT_EX source this is very easy to do.
For example, the Night Skies Project report has a page which mentions the various Acts of Congress which underpin their work; see Figure 1. The names of these acts appear in *Italics*. This was originally done by simply specifying
`\textit{Organic Act of 1916}` ...
Changing this by inventing a macro `\varepsact` the true intention is captured, at least within the L^AT_EX source. For typesetting purposes the expansion of

Advice for authors of L^AT_EX documents needing to be Tagged.




Preparing Presentations in Pandemic Times
Paulo Ney de Souza

Preparing videos for TUG 2020 (or other) presentations

- Microphones
- Network & Bandwidth
- Frame & Cameras
- Lighting
- Backgrounds
- Additional Hardware
- Recording Yourself
- Screen Capture vs. Presentation Software
- Video Editing
- Minimal requirements
- Logistics of the meeting

You have a choice of giving your lecture live, but it is much preferable for you to record it and upload it ahead of time. In this way we cover all bases in the case of a malfunction, poor bandwidth or some other problem.
We have provided below some hints about how to produce your video (live or recorded).



Advice for presenters at TUG 2020 Online.

The document at right makes use of the advice given in the preprint.

Documentation & example document

The document at left is the one submitted as a preprint for this talk. It contains a description of tagging features that can be obtained, with advice to authors on how to make the tagging process easier to achieve.


Tagging with L^AT_EX — Part 1: author considerations
Ross Moore*

Abstract
Successful tagging within PDF files generated from L^AT_EX source encourages a change in viewpoint on the nature and intent of the L^AT_EX coding. Using explicit examples from a real-world document, we illustrate how to capture such a change within the L^AT_EX source, for various structural elements. Other issues for creating archival and accessible PDF documents are discussed.

1 Introduction
With “Tagged PDF” being the accepted method for creating PDF documents enriched to satisfy Accessibility requirements [3, 6], this article is intended to address the main issues that authors and editors should be aware of, with regards to tagging and L^AT_EX usage. Examples are taken from a real-world fully-tagged research report, prepared in L^AT_EX but also employing extra coding written by the author, in a package named `tpdf` that handles the technical aspects of producing “Tagged PDF”. That research report is the one used by the author in the talk [7] at TUG 2019, and delivered remotely using Zoom

be managed to ensure long-term preservation; ...
By producing documents conforming to published standards both PDF/A [3] and PDF/UA [4], these obligations can be met in full, if not surpassed.

2 Tagging commands for special content
It is a common typographical practice to use different styling to present names of books, magazines, or publications which have a particular relevance to the topic under discussion. Certainly the fact of a different style being used indicates to a fully-sighted reader that there is a special significance, but not what that significance actually is. That has to be deduced from context. With tagging, that significance can be made explicit. And with L^AT_EX source this is very easy to do.
For example, the Night Skies Project report has a page which mentions the various Acts of Congress which underpin their work; see Figure 1. The names of these acts appear in *Italics*. This was originally done by simply specifying
`\textit{Organic Act of 1916}` ...
Changing this by inventing a macro `\varepsact` the true intention is captured, at least within the L^AT_EX source. For typesetting purposes the expansion of



Preparing Presentations in Pandemic Times
Paulo Ney de Souza

Preparing videos for TUG 2020 (or other) presentations

- Microphones
- Network & Bandwidth
- Frame & Camera
- Lighting
- Backgrounds
- Additional Hardware
- Recording Yourself
- Screen Capture vs. Presentation Software
- Video Editing
- Minimal requirements
- Logistics of the meeting

You have a choice of giving your lecture live, but it is much preferable for you to record it and upload it ahead of time. In this way we cover all bases in the case of a malfunction, poor bandwidth or some other problem.
We have provided below some hints about how to produce your video (live or recorded).

Advice for authors of L^AT_EX documents needing to be Tagged.

Advice for presenters at TUG 2020 Online.

The document at right makes use of the advice given in the preprint. We shall examine this, noting how ‘hard semantics’ are captured within the tagging.

Documentation & example document

The document at left is the one submitted as a preprint for this talk. It contains a description of tagging features that can be obtained, with advice to authors on how to make the tagging process easier to achieve.

Tagging with L^AT_EX — Part 1: author considerations

Ross Moore*

Abstract

Successful tagging within PDF files generated from L^AT_EX source encourages a change in viewpoint on the nature and intent of the L^AT_EX coding. Using explicit examples from a real-world document, we illustrate how to capture such a change within the L^AT_EX source, for various structural elements. Other issues for creating archival and accessible PDF documents are discussed.

1 Introduction

With ‘Tagged PDF’ being the accepted method for creating PDF documents enriched to satisfy Accessibility requirements [3, 6], this article is intended to address the main issues that authors and editors should be aware of, with regards to tagging and L^AT_EX usage. Examples are taken from a real-world fully-tagged research report, prepared in L^AT_EX but also employing extra coding written by the author, in a package named `tpdf` that handles the technical aspects of producing ‘Tagged PDF’. That research report is the one used by the author in the talk [7] at TUG 2019, and delivered remotely using Zoom

be managed to ensure long-term preservation; ...

By producing documents conforming to published standards both PDF/A [3] and PDF/UA [4], these obligations can be met in full, if not surpassed.

2 Tagging commands for special content

It is a common typographical practice to use different styling to present names of books, magazines, or publications which have a particular relevance to the topic under discussion. Certainly the fact of a different style being used indicates to a fully-sighted reader that there is a special significance, but not what that significance actually is. That has to be deduced from context. With tagging, that significance can be made explicit. And with L^AT_EX source this is very easy to do.

For example, the Night Skies Project report has a page which mentions the various Acts of Congress which underpin their work; see Figure 1. The names of these acts appear in *Italics*. This was originally done by simply specifying

```
\textit{Organic Act of 1916} ...
```

Changing this by inventing a macro `\varepsact` the true intention is captured, at least within the L^AT_EX source. For typesetting purposes the expansion of

Preparing Presentations in Pandemic Times

Paulo Ney de Souza

Preparing videos for TUG 2020 (or other) presentations

- Microphones
- Network & Bandwidth
- Frame & Camera
- Lighting
- Backgrounds
- Additional Hardware
- Recording Yourself
- Screen Capture vs. Presentation Software
- Video Editing
- Minimal requirements
- Logistics of the meeting

You have a choice of giving your lecture live, but it is much preferable for you to record it and upload it ahead of time. In this way we cover all bases in the case of a malfunction, poor bandwidth or some other problem.

We have provided below some hints about how to produce your video (live or recorded).



Advice for authors of L^AT_EX documents needing to be Tagged.

Advice for presenters at TUG 2020 Online.

The document at right makes use of the advice given in the preprint. We shall examine this, noting how ‘hard semantics’ are captured within the tagging. It validates for PDF/A-2a and PDF/UA, as well as passing all the automated Acrobat tests for ‘Accessibility’.

Practicalities: tagged vs. untagged

Practicalities: tagged vs. untagged

- ▶ Visual integrity:

Practicalities: tagged vs. untagged

- ▶ Visual integrity: How do we know that the Tagged version of a \LaTeX -produced PDF, looks visually identical to when processed without tagging?

Practicalities: tagged vs. untagged

- ▶ Visual integrity: How do we know that the Tagged version of a \LaTeX -produced PDF, looks visually identical to when processed without tagging?

Demonstration: using tracing-all with the `\output` routine, and check differences in the glue settings.

Practicalities: tagged vs. untagged

- ▶ Visual integrity: How do we know that the Tagged version of a \LaTeX -produced PDF, looks visually identical to when processed without tagging?

Demonstration: using tracing-all with the `\output` routine, and check differences in the glue settings.

- ▶ Page breaking; how does it affect tagging?

Practicalities: tagged vs. untagged

- ▶ Visual integrity: How do we know that the Tagged version of a \LaTeX -produced PDF, looks visually identical to when processed without tagging?

Demonstration: using tracing-all with the `\output` routine, and check differences in the glue settings.

- ▶ Page breaking; how does it affect tagging?
- ▶ Hyperlinking, both internal and external

Practicalities: tagged vs. untagged

- ▶ Visual integrity: How do we know that the Tagged version of a \LaTeX -produced PDF, looks visually identical to when processed without tagging?

Demonstration: using tracing-all with the `\output` routine, and check differences in the glue settings.

- ▶ Page breaking; how does it affect tagging?
- ▶ Hyperlinking, both internal and external: **structure destinations** with PDF 2.0

Practicalities: tagged vs. untagged

- ▶ Visual integrity: How do we know that the Tagged version of a \LaTeX -produced PDF, looks visually identical to when processed without tagging?

Demonstration: using tracing-all with the `\output` routine, and check differences in the glue settings.

- ▶ Page breaking; how does it affect tagging?
- ▶ Hyperlinking, both internal and external: **structure destinations** with PDF 2.0
- ▶ Export to XML and HTML.

Practicalities: tagged vs. untagged

- ▶ Visual integrity: How do we know that the Tagged version of a \LaTeX -produced PDF, looks visually identical to when processed without tagging?

Demonstration: using tracing-all with the `\output` routine, and check differences in the glue settings.

- ▶ Page breaking; how does it affect tagging?
- ▶ Hyperlinking, both internal and external: **structure destinations** with PDF 2.0
- ▶ Export to XML and HTML.
- ▶ ‘Beamer’ slides

Practicalities: tagged vs. untagged

- ▶ Visual integrity: How do we know that the Tagged version of a \LaTeX -produced PDF, looks visually identical to when processed without tagging?

Demonstration: using tracing-all with the `\output` routine, and check differences in the glue settings.

- ▶ Page breaking; how does it affect tagging?
- ▶ Hyperlinking, both internal and external: **structure destinations** with PDF 2.0
- ▶ Export to XML and HTML.
- ▶ ‘Beamer’ slides
- ▶ many other things ...

Accessibility, text-extraction
Accessibility, author advice for Tagged PDF
Tagging in TikZ diagrams.

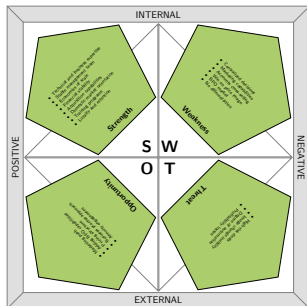
Some TikZ examples

Some TikZ examples

- ▶ A SWOT diagram⁴

Some TikZ examples

- ▶ A SWOT diagram⁴



⁴Ricardo García Fernández. 2013. TikZ SWOT diagram, <https://gist.github.com/ricardogarfe/4563453>.

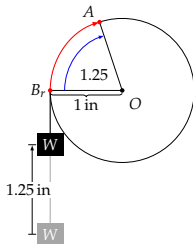
Some TikZ examples

- ▶ A SWOT diagram⁴
- ▶ Explaining how a winch works, raising

⁴Ricardo García Fernández. 2013. TikZ SWOT diagram, <https://gist.github.com/ricardogarfe/4563453>.

Some TikZ examples

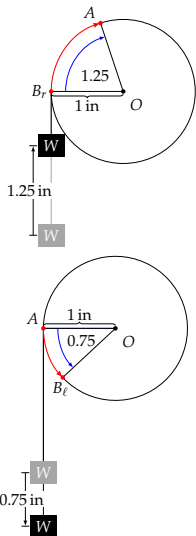
- ▶ A SWOT diagram⁴
- ▶ Explaining how a winch works, raising



⁴Ricardo García Fernández. 2013. TikZ SWOT diagram, <https://gist.github.com/ricardogarfe/4563453>.

Some TikZ examples

- ▶ A SWOT diagram⁴
- ▶ Explaining how a winch works, raising
- ▶ Explaining how a winch works, lowering



⁴Ricardo García Fernández. 2013. TikZ SWOT diagram, <https://gist.github.com/ricardogarfe/4563453>.