AcroTEX.Net

The forms16be Package

D. P. Story

Copyright © 2017 dpstory@acrotex.net
Prepared: August 30, 2017

www.acrotex.net
Version 1.2, 2017/08/28

www.AcroTeX.net

The forms16be Package

Table of Contents

1. Introduction

The `forms16be` package[1] provides support for UTF-16BE Unicode character encoding (called a big-endian character string) for the *text string type* (PDF Reference, version 1.7, beginning on page 158). Text strings are used in "text annotations, bookmark names, article threads, document information, and so forth" (to partially quote page 158). The particular application is to set property values of form fields, at least those properties that take the text strings as its value. The package contains support for Basic Latin plus the ability to enter any unicode character using the notation `\uXXXX` or `\u(XXXX)`, where 'XXXX' are four hex digits.

The code was originally designed to be used with the eforms package, but can be used with the form fields generated by hyperref, but requires some custom modification of the form field commands of hyperref.

2. Using the package

We use the example from `forms16be-ef.tex` to discuss how to use this package.

First define the unicode string to be used.

```
\defUniStr{⟨name⟩}{⟨string⟩}
```

Define a unicode string with `\defUniStr`. The ⟨*name*⟩ is the name of the unicode string you are defining, it is used later to refer to this string. The ⟨*string*⟩ argument is a combination of Basic Latin characters and unicode characters (more specifically, expressions of the form `\uXXXX` or `\u(XXXX)`, where 'XXXX' are hex digits). In the example below, we declare,

```
\defUniStr{VDV}{\u03B1 cos(\u03B8)}
\defUniStr{TU}{Don \u\EURO Story "\u03B1 cos(\u03B8)"}
```

The definition file `uni4basic-latin.def` defines the encoding for the Basic Latin character set plus a few more definitions, including `\EURO`. See that file for more details.

After declaring and naming your unicode strings, use `\unicodeStr` to set the values of selected field properties.

```
\unicodeStr{⟨name⟩}
```

`\unicodeStr{⟨name⟩}` expands to the ⟨*string*⟩ argument associated with ⟨*name*⟩, as declared by the `\defUniStr` command above. For example, `\unicodeStr(VDV)`, declared above, expands to

```
FEFF03B100200063006F0073002803B80029
```

Notice the first four hex digits, 'FEFF', these are 'marker digits'. As the PDF Specification describes, the markers are used to signal the beginning of a big-endian hex string.

For eforms, incorporating these ideas into field creation is easy:

[1] The code in this package was extracted from aeb_pro. The code itself did not depend on PostScript, it is useful, therefore, to remove it from aeb_pro code base and make it available to LaTeX users with different workflows (pdflatex, lualatex, and xelatex).

```
\textField[\TU{\unicodeStr(TU)}
  \DV{\unicodeStr(VDV)}\V{\unicodeStr(VDV)}][tst16be]{1.5in}{11bp}
```

The result is α cos(θ). Pass your mouse over the field to see the tool tip as well.

Within the argument of ⟨*string*⟩, backslash, left and right braces are not defined in the `uni4basic-latin.def` file. They, therefore, cannot appear as literals within ⟨*string*⟩. Should you need these characters, use the following:

- `\u005C` or `\u\BSLASH` for backslash (\).
- `\u007B` or `\u\LBRACE` for left brace ({);
- `\u007D` or `\u\RBRACE` for right brace (});

(Other definitions within the `uni4basic-latin.def` file are `\EURO` and `\DQUOTE`.) The introduction of the command versions of unicode brings up another problem, that of obeying spaces.

Suppose you wanted to initialize a field property with '`\LaTeX`'. To obtain this value we would type '`\u\BSLASH LaTeX`'. But, because ⟨*string*⟩ is under the influence of `\obeyspaces`, the specified initialization appears as '`\ LaTeX`', that is, there is a space that follows the backslash; of course, we cannot specify `\u\BSLASHLaTeX` as that would get an undefined command error. The solution is to enclose `\BSLASH` in parentheses; if we type `\u(\BSLASH)LaTeX` we obtain the desired result:

```
\defUniStr{LaTeX}{\u(\BSLASH)LaTeX}
\textField[\V{\unicodeStr(LaTeX)}
  \DV{\unicodeStr(LaTeX)}][tstLaTeX]{1in}{11bp}
```

The above code results in \LaTeX.

3. List of field properties that take a text string

The property entries in a form field that support the text string type are **DV**, **V**, **TU**, **CA**, **RC**, and **AC**. The eforms key counterparts are `\DV`, `\V`, `\TU`, `\CA`, `\RC`, and `\AC`. When the argument of any of these begins with `\unicodeStr`, eforms detects this and passes its argument to the unicode keys `\uDV`, `\uV`, `\uTU`, `\uCA`, `\uRC`, and `\uAC`. Normally, the value of **DV**, for example, is `DV (`⟨*text*⟩`)`, its value is enclosed in parentheses; when the value consists of hex digits, angle brackets are required, like so `DV <`⟨*string*⟩`>`.

If you enter raw big-endian hex digits, use the special `\u`⟨*cmd*⟩ version of the keys, for example,

```
\pushButton[\CA{\unicodeStr(263A263C)}][btn]{.5in}{11bp}
\pushButton[\uCA{FEFF263A263C}][btn]{.5in}{11bp}
```

expands to ☺☼ or ☺☼. The first version uses `\unicodeStr` with raw hex digits, because `\unicodeStr` is used, the unicode is detected and `\uCA` is used (with angle brackets). In the second example, `\unicodeStr` is not used, so `\uCA` must be explicitly used; also, the unicode marker `FEFF` must explicitly appear as well. (`\unicodeStr` automatically inserts the marker.) The results of the markup forms is the same.

4. Fonts

When using unicode to reference glyphs, such as the dings presented above, it is important that the fonts the PDF viewer uses contains the glyphs. If the glyph does not exist in the font, the viewer might be successful at substituting the font. The viewer is not always successful. For example,

```
\defUniStr{subS}{x\u209B}
\textField[\V{\unicodeStr(subS)}
  \DV{\unicodeStr(subS)}][tstsubS}{.5in}{11bp}
```

The result is x•, the default value of this field should be x_S, is it so? The default font is `Helv`. Now, if we change to the `TiRo` font, we obtain xₛ. A better result! The (newer) `\u209B` glyph is not supported for all fonts. Generally, you'll have to find a font that works and is available to the end user as well.

5. A combobox example

For combo boxes and list boxes, things are slightly more complicated. In the example below, we define a combo box. First, define the appearance values of the combo box (the string that is seen listed in the combo box).

```
\defUniStr{myEuro}{\u20AC (Euro)}
\defUniStr{myYen}{\u00A5 (Yen)}
\defUniStr{mySheqel}{\u20AA (Sheqel)}
\defUniStr{myPound}{\u00A3 (Pound)}
\defUniStr{myFranc}{\u20A3 (Franc)}
```

Then we can define our combo box. According to the PDF file format, unicode strings should be enclosed in angle brackets <XXXXXXXXXXXX>. In initializing the combo box below, the `\unicodeStr` command is used, but this time it is enclosed in angle brackets. (That is all the "u" versions of the keys do above is to automatically insert the angle brackets for you. Here we have to do it ourselves. (I suppose one could have a helper command, but you can handle it.)

```
\comboBox[\Ff\FfEdit\DV{Euro}\V{Euro}\textFont{Arial}
\BG{0.98 0.92 0.73}\BC{0 .6 0}][{myCombo}{1in}{11bp}
{*{[(Euro)<\unicodeStr(myEuro)>]%
    [(Yen)<\unicodeStr(myYen)>]%
    [(Sheqel)<\unicodeStr(mySheqel)>]%
    [(Pound)<\unicodeStr(myPound)>]%
    [(Franc)<\unicodeStr(myFranc)>]}
}
```

where the * in the position shown above is a token that signals the passing of a raw form of the value options of a combo or list box; it is defined in `eforms`.

The result is € (Euro). Very swave!

6. Demonstration files

There are three sample files:

- `forms16b3-ef.tex` uses the eforms package to create form fields and demonstrate how to initialize form fields using unicode.
- `forms16b3-hy.tex` uses form fields created by hyperref. hyperref does not support initializing the value of a field, for example, with big-endian, so a redefinition is needed.
- `forms16b3-ap.tex` demonstrates the forms16be with aeb_pro (dvips/Distiller workflow required). For consistency with previously documented behavior of aeb_pro, the `linktoattachments` option is used to import forms16be. It is not necessary, however, to use `linktoattachments`.

7. My retirement

Now, I simply must get back to it. DPS