

DESCRIPTION

App::FontSample creates PDF font specimens, font comparisons, compact font collections, and character charts.

App::FontSample does not provide a font collection. Library callers supply one or more PDF::Content::FontObj objects. The installed font-sample program can load an OTF or TTF file directly.

The current release supports Letter and A4 paper, portrait and landscape orientation, configurable margins, JSON job files, registered language pangrams, and four layouts (click each link to view on GitHub):

- specimen
- comparison
- collection
- characters

INSTALLATION

```
zef install App::FontSample
```

FIRST SAMPLE

The easiest first test uses the standard PDF core font Times-Roman, so no external font file is required.

```
use PDF::API6;
use App::FontSample;

my $pdf = PDF::API6.new;
my $font = $pdf.core-font :family<Times-Roman>;

create-font-sample(
    $font,
    :name<Times-Roman>,
    :title<Font Specimen>,
    :language<en>,
    :output<sample.pdf>,
);
```

The specimen identifies the font, labels the alphabet and punctuation sizes, labels the pangram as English (en), and includes a size waterfall with each point size shown at the left.

CURRENT LAYOUTS

specimen

The specimen layout produces one specimen page per font. It shows the font name, alphabet, numerals and punctuation, a labeled language pangram, and a size waterfall.

```
create-font-sample(
    $font,
    :name<Times-Roman>,
    :layout<specimen>,
    :title<Font Specimen>,
    :language<en>,
    :output<specimen.pdf>,
);
```

See an example at [click here](#) .

An explicit pangram must also identify its language so the PDF can label it:

```
create-font-sample(  
$font,  
:name<Times-Roman>,  
:layout<specimen>,  
:language<en>,  
:pangram('Pack my box with five dozen liquor jugs.'),  
:output<specimen-custom.pdf>,  
);
```

NOTE

The collection and comparison sections below render each font at the same nominal point size. Visible character heights may differ because each typeface has its own font metrics.

comparison

The comparison layout displays the same text in several fonts at one stated point size. The font name is shown above each sample. The layout continues onto additional pages when needed.

```
create-font-comparison-sample(  
@entries,  
:layout<comparison>,  
:title<Font Comparison>,  
:comparison-size(18),  
:text(  
'The quick brown fox jumps over the lazy dog. 0123456789'  
),  
:output<comparison.pdf>,  
);
```

collection

The collection layout is a compact one-page inventory of fonts. Each font name is followed by a short typographic identification string. The page title states the sample point size.

```
create-font-collection-sample(  
@entries,  
:layout<collection>,  
:title<Font Collection>,  
:sample-size(14),  
:text('Aa Bb Cc Dd Ee Ff Gg 0123456789'),  
:output<collection.pdf>,  
);
```

If the requested collection cannot fit within the page margins, the layout reports an error rather than overflowing the page.

characters

The characters layout displays selected characters in a grid. The page heading identifies the font and glyph point size, and each cell labels the glyph with its Unicode code point.

```
create-font-sample(  
$font,  
:name<Times-Roman>,  
:layout<characters>,  
:title<Character Sample>,  
:characters(  
'ABCDEFGHijklmnopqrstuvwxyz0123456789'
```

```

),
:columns(8),
:glyph-size(30),
:output<characters.pdf>,
);

```

Library callers may supply characters as a string or as a positional collection containing one-character strings, integer Unicode code points, or strings such as U+0041.

FONT COLLECTIONS

A comparison or collection uses `App::FontSample::FontEntry` objects.

```

use App::FontSample;
use App::FontSample::FontEntry;

my @entries;

@entries.push: App::FontSample::FontEntry.new(
: name<Times-Roman>,
: font($times),
);

@entries.push: App::FontSample::FontEntry.new(
: name<Helvetica>,
: font($helvetica),
);

create-font-collection-sample(
@entries,
: layout<comparison>,
: comparison-size(18),
: output<comparison.pdf>,
);

```

PAPER AND PAGE OPTIONS

Both Letter and A4 are supported. Pages may be portrait or landscape, and the margin may be set in PDF points.

```

create-font-sample(
$font,
: name<Times-Roman>,
: paper<A4>,
: landscape,
: margin(36),
: language<en>,
: output<a4-landscape.pdf>,
);

```

There are 72 PDF points in one inch.

LANGUAGE PANGRAMS

The current built-in pangram registry includes English under the two-letter code `en`.

```

create-font-sample(
$font,
: name<Times-Roman>,
: language<en>,
: output<english.pdf>,
);

```

The resulting specimen labels the text English (en) pangram.

List the currently registered language codes with:

```
font-sample --languages
```

Applications may register additional pangrams and language names at runtime.

OTF AND TTF FILES

The installed font-sample program loads an OTF or TTF file directly:

```
font-sample \  
--font=/path/to/font.otf \  
--output=sample.pdf \  
--layout=specimen \  
--language=en
```

Current direct-mode layout names are:

```
specimen  
comparison  
collection  
characters
```

JSON INPUT

font-sample can read one or more jobs from a JSON definition file:

```
font-sample --config=font-samples.json
```

For example:

```
{  
  "output": "output/sample.pdf",  
  "layout": "specimen",  
  "language": "en",  
  "fonts": [  
    {  
      "file": "fonts/MyFont.otf",  
      "name": "My Font"  
    }  
  ]  
}
```

REPRODUCIBLE PDF OUTPUT

For generated examples or other PDFs that should be byte-for-byte reproducible, use :reproducible from Raku or --reproducible in direct command-line mode. JSON jobs may use "reproducible": true. Normal PDF generation retains the default unique document identification and metadata behavior.

OPTIONAL NOTOFONTS-OT USE

NotoFonts-OT is not a dependency of App::FontSample. It is one optional source of PDF::Content::FontObj objects.

```
use NotoFonts-OT;  
use App::FontSample;  
use App::FontSample::FontEntry;  
  
my @entries;
```

```

for <
NotoSerif-Regular
NotoSerif-Bold
NotoSans-Regular
NotoSans-Bold
> -> $code {
my $font = get-loaded-font($code);

@entries.push: App::FontSample::FontEntry.new(
:name($code),
:$font,
);
}

create-font-collection-sample(
@entries,
:layout<comparison>,
:comparison-size(18),
:output<noto-comparison.pdf>,
);

```

EXAMPLES

examples/current-capabilities.raku uses PDF core fonts and creates four PDFs, one for each current layout:

- example-specimen.pdf
- example-comparison.pdf
- example-collection.pdf
- example-characters.pdf

Run the examples like this:

```
raku -Ilib examples/current-capabilities.raku
```

TESTING

Run the normal test suite with `mi6 test` (for modules under `App::Mi6` management) or `zef ..`

```
mi6 test
```

The normal test suite does not require NotoFonts-OT.

FUTURE WORK

This README describes current capability only. Planned additions and changes are maintained in `docs/TODO.rakudoc`.

SOURCE

The project source is maintained in the `tbrowder/App-FontSample` repository on GitHub.

AUTHOR

Tom Browder <tbrowder@acm.org>

COPYRIGHT AND LICENSE

© 2026 Tom Browder

This library is free software; you may redistribute it or modify it under the Artistic License 2.0.