

Installing the TX/PX fonts for SWP5/pdfTeX

Philip A. Viton

October 21, 2004

1 Introduction

This note explains how you can use Type1 fonts in an SWP5+ document — but only when producing pdf. I do this by way of two examples, both using fonts created by Young Ryu: the TX fonts, which are an alternative to Times+Mathtime, and the PX fonts, which provide a Palatino-like setup. SWP already includes support (Mathtime) for Times typesetting; however, the TX fonts may be of interest since, unlike Mathtime, you get bold Greek in Times-like appearance; a “real” caps-and-small-caps font; and a “slanted Times” font, which is completely missing from mathtime. The PX fonts do all this, but in Palatino. Taken together, the two illustrate some of interesting effects we can get in SWP5, now that PDF processing is supported directly; and also how relatively easy it is to set them up.

Important Notes. This is for SWP5 or later, which comes with direct pdfTeX support. Earlier SWP versions are not officially supported, and may or may not work. Note also that this gives you alternative font support for pdfTeX only, and not for DVI (ordinary typeset output). Under recent versions of Windows (Win2K, WinXP, possibly also NT4) you should be able to install Type1 (pfb) fonts directly. However, for SWP users the issue is whether the TrueTeX previewer will be able to work with them. I’ve had varying results here, so I’m not even going to try and suggest how it could be done.

Here are some non-math illustrations:

- PDF version of this document, using SWP5’s standard Times Fonts: [txfontsSWP-Times.pdf](#)
- PDF version of this document, using TX Times fonts (URW Times): [txfontsSWP-TX.pdf](#)
- PDF version of this document, using PX Palatino Fonts: [txfontsSWP-PX.pdf](#)

2 Installing the TX fonts

Young Ryu’s TX fonts support Times-like typesetting, including a set of specially designed fonts for math.

Here are step-by-step instructions for installing the TX fonts, which should take perhaps 10 minutes after you get the archive. Note that the font metrics for the TX fonts differ from those of mathtime, so you’re not guaranteed that documents processed with mathtime and with the TX fonts will look exactly the same.

1. Get the support from CTAN. The location is in `fonts/txfonts`. Once you’re there, you should be able to retrieve the entire directory (and all subdirectories) as a zip file (about 5.6MB). Alternatively, if you’re a member of TUG, `txfonts.zip` is on the CTAN CD number 2.

The two methods give you different upper-level directory structures, but that’s not important, because we’ll be installing (some) of the files to our own special directories. Therefore, *you should turn off any facility in your unzipper that restores the directory structure*. For example, in WinZip, make sure that “Use Folder Names” is unchecked.

2. Create the following folder:

- `c:\swp50\TCITex\TeX\latex\contrib\supported\txfonts`

3. This step is not strictly necessary, but will keep (most of) the new files separate from those supplied with SWP5, which is probably desirable. Create the following directories:

- `c:\swp50\TCITex\fonts\tfm\txfonts`
- `c:\swp50\TCITex\fonts\vf\txfonts`

- `c:\swp50\TCITeX\fonts\type1\txfonts`

4. Unzip the files to the indicated directories (where the part of destination in brackets refers to the folders you created in the previous step, and will of course be missing if you decided to lump everything together):

Archive Files	Destination
<code>..\txfonts\tfm*.tfm</code>	<code>c:\swp50\TCITeX\fonts\tfm\txfonts\</code>
<code>..\txfonts\tfm*.vf</code>	<code>c:\swp50\TCITeX\fonts\vf\txfonts\</code>
<code>..\txfonts\tfm*.pfb</code>	<code>c:\swp50\TCITeX\fonts\type1\txfonts\</code>
<code>..\txfonts\input*.*</code>	<code>c:\swp50\TCITeX\TeX\latex\contrib\supported\txfonts</code>
<code>..\txfonts\dvips\tx8r.enc</code>	<code>c:\swp50\TCITeX\fonts\type1\txfonts\</code>
<code>..\txfonts\dvips\txr2.map</code>	<code>c:\swp50\TCITeX\pdftex\config\</code>

5. Now open the included archive file `tx_patch1.zip` and unzip as follows, overwriting existing files:

Archive File	Destination
<code>txmi.tfm</code>	<code>c:\swp50\TCITeX\fonts\tfm\txfonts\</code>
<code>txmi1.tfm</code>	<code>c:\swp50\TCITeX\fonts\tfm\txfonts\</code>
<code>txmi.vf</code>	<code>c:\swp50\TCITeX\fonts\vf\txfonts\</code>
<code>txmi1.vf</code>	<code>c:\swp50\TCITeX\fonts\vf\txfonts\</code>

6. Rename or copy `txr2.map` as `txr3.map`.
7. Go to `c:\swp50\tcitex\pdftex\config` and open `pdftex.cfg` in a text editor. At the end of the file insert the following line:

```
map +txr3.map
```

Close and save `pdftex.cfg`.

8. You must now make a few changes in the new `txr3.map` file, to get the “Times” fonts provided by the SWP distribution. Open `txr3.map` in a text editor. You must first change the names of the first 6 pfb fonts as follows:

TX name	PFB file
<code>rtxptmb</code>	<code>utmb8a.pfb</code>
<code>rtxptmbo</code>	<code>utmb8a.pfb</code>
<code>rtxptmbi</code>	<code>utmbi8a.pfb</code>
<code>rtxptxr</code>	<code>utmr8a.pfb</code>
<code>rtxptxro</code>	<code>utmr8a.pfb</code>
<code>rtxptxri</code>	<code>utmri8a.pfb</code>

where “TX Name” is the identifier at the far left of the line in the map file, “PFB file” is the name that replaces the existing `.pfb` font name.

Then you must made another set of four changes in the section of the file referring to “URW NimbusSanL”. These changes are as follows:

TX name	PFB file
<code>rtxphvr</code>	<code>uhvr8a.pfb</code>
<code>rtxphvro</code>	<code>uhvr8a.pfb</code>
<code>rtxphvb</code>	<code>uhvb8a.pfb</code>
<code>rtxphvbo</code>	<code>uhvb8a.pfb</code>

Remember not to omit the “<” preceding each of the pfb file names: this is what ensures that the fonts get included in your PDF file.

That’s it.

To use the new facility, you need to load the `txfonts` package. However, there’s one complication: there appears to be a small inconsistency between `tcilatex.tex` — loaded by all non-portable SWP documents — and the `txfonts` package. Therefore you must load `txfonts` *after* the line `\input{tcilatex}` in the preamble. There may be a better way to handle this, so if anyone has an idea, let me know. Alternatively, the MSI folks might take a look and see if they can spot the problem. See the last section for a package to help with this.

3 Installing the PX fonts

Young Ryu has also produced support for math typesetting in Palatino rather than Times: these are the PX fonts.

Here’s how to set them up: again, note that this is for PDF production only.

1. **Important** : the PX fonts require that you also install the TX fonts, since they have some files in common, and those files are not repeated in the PX fonts archive. In particular, you need `tx8r.enc` (and there is no equivalent in the PX fonts distribution). So if you’ve not done so, install the TX fonts, per the instructions above.
2. Obtain the PX font support. This is available on CTAN, in `fonts/pxfonts`. The procedure for getting the archive is the same as for obtaining the TX fonts;

the zip file is about 4.2MB (and is also available on the CTAN CD from TUG). Note that the next couple of steps are the same as for installing the TX fonts, with “tx” replaced by “px”.

3. Create the following folder:

- `c:\swp50\TCITeX\TeX\latex\contrib\supported\pxfonts`

4. This step is not strictly necessary, but will keep (most of) the new files separate from those supplied with SWP5, which is probably desirable. Create the following directories:

- `c:\swp50\TCITeX\fonts\tfm\pxfonts`
- `c:\swp50\TCITeX\fonts\vf\pxfonts`
- `c:\swp50\TCITeX\fonts\type1\pxfonts`

5. Unzip these files to the indicated directories (where the part of destination in brackets refers to the folders you created in the previous step, and will of course be missing if you decided to lump everything together):

Archive Files	Destination
<code>..\pxfonts\tfm*.tfm</code>	<code>c:\swp50\TCITeX\fonts\tfm\pxfonts\</code>
<code>..\pxfonts\tfm*.vf</code>	<code>c:\swp50\TCITeX\fonts\vf\pxfonts\</code>
<code>..\pxfonts\tfm*.pfb</code>	<code>c:\swp50\TCITeX\fonts\type1\pxfonts\</code>
<code>..\pxfonts\input*.*</code>	<code>c:\swp50\TCITeX\TeX\latex\contrib\supported\pxfonts</code>
<code>..\pxfonts\dvips\pxr2.map</code>	<code>c:\swp50\TCITeX\pdftex\config\</code>

6. Now open the included archive file `px_patch1.zip` and unzip as follows, overwriting existing files:

Archive File	Destination
<code>pxmi.tfm</code>	<code>c:\swp50\TCITeX\fonts\tfm\pxfonts\</code>
<code>pxmi1.tfm</code>	<code>c:\swp50\TCITeX\fonts\tfm\pxfonts\</code>
<code>pxmi.vf</code>	<code>c:\swp50\TCITeX\fonts\vf\pxfonts\</code>
<code>pxmi1.vf</code>	<code>c:\swp50\TCITeX\fonts\vf\pxfonts\</code>

7. Rename or copy `pxr2.map` as `pxr3.map`.
8. Go to `c:\swp50\tcitex\pdftex\config` and open `pdftex.cfg` in a text editor. At the end of the file insert the following line:

```
map +pxr3.map
```

Close and save `pdftex.cfg`.

9. You must now make a few changes in the new `pxr3.map` file, to get the “Palatino” fonts provided by the SWP distribution. Open `pxr3.map` in a text editor. You must change the names of the first 6 pfb fonts as follows:

PX name	PFB file
<code>rpxpplb</code>	<code>uplb8a.pfb</code>
<code>rpxpplbo</code>	<code>uplb8a.pfb</code>
<code>rpxpplbi</code>	<code>uplbi8a.pfb</code>
<code>rpxpplr</code>	<code>uplr8a.pfb</code>
<code>rpxpplro</code>	<code>uplr8a.pfb</code>
<code>rpxpplri</code>	<code>uplri8a.pfb</code>

where “PX Name” is the identifier at the far left of the line in the map file, “PFB file” is the name that replaces the existing `.pfb` font name. Remember not to omit the “<” preceding each of the pfb file names: this is what ensures that the fonts get included in your PDF file.

10. To use the PX fonts, just include the `pxfonts` package, eg via `\RequirePackage{pxfonts}` in the preamble. Just as with the TX fonts, this must be loaded *after* `\input{tcilatex}`. See below for a package supporting this, together with a sketch of how to customize the PDF output.

Note that using the PX fonts is a bit more problematic than the TX fonts, because DVI production has no facilities at all for Palatino. This means that documents produced using these Palatino fonts will be compile-able *only* through pdfTeX.

4 A package

The following sketch of a PDF-helper package is designed to load the TX or PX fonts only when you are producing pdf output. (If you’re producing regular DVI output, it does nothing). It also loads the `hyperref` package — which produces “live” links in your pdf — and illustrates how you might set up some default customizations for the PDF output. I call this package `pavpdf.sty`. You should load it in the preamble *after* `\input{tcilatex}` via a `\RequirePackage` line. It takes two non-required options: `tx` for the TX fonts or `px` for the PX fonts. So to get a PDF version of your work using the PX fonts you’d say `\RequirePackage[px]{pavpdf}`. Here’s the code.

```

% ***** pavpdf.sty *****

\newcommand{\@pavtxfonts}{0}
\newcommand{\@pavpxfonts}{0}

\DeclareOption{tx}{
  \renewcommand{\@pavtxfonts}{1}
}
\DeclareOption{px}{
  \renewcommand{\@pavpxfonts}{1}
}

\ProcessOptions

\ifx\pdfoutput\undefined \endinput
\else
  \ifcase\pdfoutput \endinput
  \else
% we set up hyperref to do no border around links
  \RequirePackage[pdftex,colorlinks,
    bookmarks=false,
    pdfstartview=XYZ,
  ]{hyperref}
  \def\msihyperref#1#2#3#4{#2\href{#4}{#1}#3}
  \ifnum\@pavtxfonts=1
    \typeout{Using the TX fonts ...}
    \RequirePackage{txfonts}
  \fi
  \ifnum\@pavpxfonts=1
    \typeout{Using the PX fonts ...}
    \RequirePackage{pxfonts}
  \fi
  \fi
\fi

\endinput

```

Commentary: there are three cases that need to be tested for. If `\pdfoutput` is undefined, then we're producing DVI, and we exit (`\endinput`). If `\pdfoutput` is defined, but is zero, then we're also producing DVI, and we again exit. Otherwise, `\pdfoutput` is defined and greater than 0, and we really are producing PDF. In this case we begin by loading the `hyperref` package, with the `pdftex`-specific settings. The option `colorlinks` does what you'd expect: produces live hyperlinks in color. (You can adjust the colors too, but I don't try that here). The option `bookmarks=false` turns off the initial display of bookmarks, a display which I find less than helpful. Note that the bookmarks are still there, they're just not displayed when the document loads. The final option `pdfstartview=XYZ` ensures that the width of the document is adjusted to fit into the user's window. Again, this is a personal preference of mine: you might prefer another view. The `hyperref` manual (and *The LaTeX Web Companion*) documents the available options.

Next we re-define the `\msihyperref` macro to produce an `href` in the document. I'm not sure that this definition is optimal — what gets displayed is made up of all three of the “before-text”, the screen text, and the “after text” — but if you don't like it, it's easy to change. I'm also not sure that this will work for all `msihyperref`s: it's tailored to the way I've been using them.

Finally we load the `TX` or `PX` font package, depending on whether you've provided an optional argument or not. (Note that we don't check that you haven't provided *both* optional arguments: if you do, then you'll probably get the `PX` fonts, but things may get out of hand).

5 Concluding Remarks

The catch in all this, or course, is the disconnect it introduces between PDF and DVI. It would be very nice if the `TrueTeX` previewer could be made to handle `Type1` fonts reliably (either using Windows native facilities, or via the free `ATM-Lite` system). In my view, integrating `Type1` fonts should be considered an urgent priority for `TrueTeX` development.