romandeadf

Clea F. Rees*

v2.1 (SVN Rev: 10366) 2024/09/18

Abstract

Hirwen Harendal, Arkandis Digital Foundry (ADF) has produced the Romande ADF font collection. This guide outlines the TEX/LATEX support provided by romandeadf for version 1.008 of the fonts.

Contents

*Bug tracker: codeberg.org/cfr/nfssext/issues | Code: codeberg.org/cfr/nfssext | Mirror: github.com/cfr42/nfssext

1 Introduction

This document explains how to use the TeX/LaTeX support provided for version 1.008 of the Romande ADF font collection developed by Hirwen Harendal of the Arkandis Digital Foundry (ADF). romandeadf includes copies of the fonts in postscript type 1 format. Further information about the fonts themselves and alternative font formats for use with other programmes can be found at http://arkandis.tuxfamily.org/adffonts.html. The fonts are released under the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version, with font exception. For details, see NOTICE.txt and COPYING.

The TeX/LaTeX support package consists of all files listed in `manifest.txt` and these files are released under the LaTeX Project Public Licence as explained in the included licensing notices. Please file bugs for any problems so that I can solve them if I can. If you can correct the problems and include the fix, that would be even better.

Table 1: RomandADF original and Berry font names.

TEX directory	font families	Original name	TEX name
romande	yrdw, yrdaw	RomandeADFScriptStd-Italic	yrdriw8a
	yrd, yrda	RomandeADFStd-DemiBold	yrdd8a
		RomandeADFStd-DemiBoldItalic	yrddi8a
		RomandeADFStd-Italic	yrdri8a
		RomandeADFStd-Regular	yrdr8a
		RomandeADFStyleStd-DemiBold	yrddc8a
		RomandeADFStyleStd-Regular	yrdrc8a

2 The collection

Romande ADF is a serif family with oldstyle figures designed as a substitute for Times, Tiffany or Caslon. The family currently includes upright, italic and small-caps shapes in each of regular and demi-bold[1] weights and an italic script in regular. The support package renames the fonts according to the Karl Berry fontname scheme and defines four families. Two of these primarily provide access to the 'standard' or default characters while the 'alternate' families support alternate characters, additional ligatures and the long s[2]. The included package files provide access to these features in LATEX as explained in section 4 on the next page and section 5 on page 5.

Table 1 lists the original and Berry names for all included fonts.

3 Requirements

Apart from such obvious requirements as LATEX 2$_\varepsilon$, the LATEX support provided by `romande.sty` requires `nfssext-cfr`. Without this, you will get errors complaining that the package cannot be found and you will not be able to use any of the additional font commands described in section 5 on page 5.

The documentation requires in additional packages. These are all standard and available from CTAN but you can always comment out the relevant lines in `romandeadf.tex` if you wish.

[1] As `romande` does not include bold or bold extended fonts, `romadeadf` substitutes demi-bold so that the ordinary commands to select bold work as expected.

[2] Section 4.1 on the following page describes the encodings used to create these families. For further details see the encoding files `t1-romandeadf.etx` and `t1-romandeadf-alt.etx`.

4 The support package

4.1 Encodings

The package supports modified EC/T1 and Text Companion (TS1) encodings. Most characters in the EC encoding are available and the fonts provide a small number of characters from the TS1 encoding as well, including the €. The regular version of the EC/T1 encoding (`t1-romandeadf.enc`) reassigns one slot which would otherwise be empty due to missing glyphs which `fontinst` cannot fake. In the T1 encoding, this slot is standardly used for the per thousand zero. `t1-romandeadf.enc` uses the slot for the slashed zero (Ø). The fonts also lack the unfakable Sami Eng/eng characters (Ŋ/ŋ). Although these slots remain in the regular version of the encoding, they are empty due to the lack of suitable glyphs.

The 'alternative' version of the EC/T1 encoding (`t1-romandeadf-alt.enc`) provides access to the full range of ligatures available — including 'ct', 'sp', 'st' and 'ft' — in addition to the alternate 'Q' (Q) and the long s (ſ). The slots used by T1 for the Sami characters and per thousand zero are reassigned first since these would otherwise be empty, but because further slots are required, a number of characters normally available in the EC encoding are unavailable. These are Ohungarumlaut (Ő), ohungarumlaut (ő), Uhungarumlaut (Ű) and dbar (đ). Attempting to access these characters while using the alternative versions of the fonts may result in errors of various kinds and will certainly produce unexpected output even though the characters are provided by the fonts, as the previous sentence demonstrates. To access these glyphs, ensure that the regular versions of the fonts are active.

4.2 LaTeX package

To use the fonts in a LaTeX document, add `\usepackage{romande}` to your document preamble. This will set the default serif/roman family to `yrd` (romande) and enable access to the various alternates and additional glyphs available in the other families.

`alt` (*opt.*) = `true|false`

Loading `romande` with this option will select the 'alternative' version as the default serif/roman family. ***This option is not recommended unless you are*** **certain** ***you do not wish to access any of the characters described in section 4.1.*** You should also note that this option will mean all of the additional ligatures will be active, which may not be what you want.

Initially `false`; defaults to `true` if used with no value.

`scale` (*opt.*) = ⟨*scaling factor*⟩

Scale all fonts provided by the package by ⟨*scaling factor*⟩, which should be a positive integer or simple decimal such as `2` or `1.2`. This option is intended for cases where

Table 2: RomandeADF styles.

style	style command	text command	effect
alternate	`\altstyle`	`\textalt{}`	'alternative'
script/swash	`\swashstyle`	`\textswash{}`	italic, regular script

the fonts should be scaled to match other families used in the document e.g. for consistency with the size of sans-serif or typewriter fonts.

Initially empty, which is equivalent to `1` but more efficient.

Note that loading `romande.sty` will not affect the default sans-serif or typewriter families.

5 Additional font selection commands

The LaTeX package romande loads nfssext-cfr which is an extension of the package nfssext supplied by Philipp Lehman as part of The Font Installation Guide. The file extends the font selection commands to facilitate access to various font features. Both the original and the extension are designed for use with a wide range of fonts. For this reason, only a subset of the additional commands are relevant to any particular font support package. Those relevant to romandeadf are described below.

5.1 nfssext-cfr

These commands are available when romande is loaded. If for some reason you wish to make them available when no relevant package is loaded, use `\usepackage{nfssext-cfr}` in your document preamble.

5.1.1 Styles

Alternate and swash styles may be accessed using the macros shown in table 2.

`\textswash` `{`⟨*text*⟩`}`

`\swashstyle` `\swashstyle` and `\textswash{}` switch to the script font (yrdw/yrdaw) regardless of the current shape and weight — you do not need to select italic shape or regular weight.

`\textalt` `{`⟨*text*⟩`}`

`\altstyle` `\altstyle` and `\textalt{}` switch to the 'alternative' families (yrda/yrdaw). Within the scope of these commands and except for small-caps:

- `Q*` will typeset the alternate ‘Q’ (Q);
- `s*` will typeset the long s (ſ);
- `ct`, `sp`, `st` and `s*t` will typeset the corresponding ligature (ct/ſp/ſt/ſt);
- attempting to typeset certain standard characters will produce unexpected results (see section 4.1 on page 4).

To make things slightly more convenient, `Q*` and `s*` will not be typeset literally when the regular encoding (or small-caps) is active. Rather, these sequences will simply typeset the standard ‘Q’ and ‘s’.

For example, suppose that romande was loaded and the following commands set up:

```
\newcommand{\fytext}{%
  Sphinx of black quartz, judge my vow.\par
  Querulous sponges act last.\par
  Q*uerulous s*ponges* act las*t.}
\newcommand{\fytest}{%
  \fytext\smallskip\par
  {\itshape\fytext}\smallskip\par
  {\scshape\fytext}\smallskip\par
  {\bfseries\fytext\smallskip\par
  {\itshape\fytext}\smallskip\par
  {\scshape\fytext}}\smallskip\par
  {\swashstyle\fytext}}
```

Then:

```
--- ‘regular’ ---\smallskip\par
\fytest\bigskip\par
--- ‘alternate’ ---\smallskip\par
\altstyle
\fytest
```

produces:

— ‘regular’ —

Sphinx of black quartz, judge my vow.

Querulous sponges act last.

Querulous sponges act last.

Sphinx of black quartz, judge my vow.

Querulous sponges act last.

Querulous sponges act last.

Sphinx of black quartz, judge my vow.

Querulous sponges act last.

Querulous sponges act last.

Sphinx of black quartz, judge my vow.

Querulous sponges act last.

Querulous sponges act last.

Sphinx of black quartz, judge my vow.

Querulous sponges act last.

Querulous sponges act last.

Sphinx of black quartz, judge my vow.

Querulous sponges act last.

Querulous sponges act last.

Sphinx of black quartz, judge my vow.

Querulous sponges act last.

Querulous sponges act last.

— ‘alternate’ —

Sphinx of black quartz, judge my vow.

Querulous ſponges act laſt.

Querulous ſpongeſ act laſt.

Sphinx of black quartz, judge my vow.

Querulous ſponges act laſt.

Querulous ſpongeſ act laſt.

Sphinx of black quartz, judge my vow.

Querulous sponges act last.

Querulous sponges act last.

Sphinx of black quartz, judge my vow.

Querulous ſponges act laſt.

Querulous ſpongeſ act laſt.

Sphinx of black quartz, judge my vow.

Querulous ſponges act laſt.

Querulous ſpongeſ act laſt.

SPHINX OF BLACK QUARTZ, JUDGE MY VOW.

QUERULOUS SPONGES ACT LAST.

QUERULOUS SPONGES ACT LAST.

Sphinx of black quartz, judge my vow.

Querulous ſponges act laſt.

Querulous ſpongeſ act laſt.

5.1.2 The slashed zero (ø)

Both of the modified T1 encodings used by romandeadf include a non-standard ligature to accommodate the slashed zero. Provided romande is loaded and the default serif/roman family is active, `0*` will produce the slashed zero (ø).

A Installation

The vast majority of users should IGNORE this section entirely. romandadf is included in all major TEX distributions and should be installed as part of your TEX installation. Installing the package yourself should be done only as a last resort or an educational exercise.

Note, in particular, that this version of romandadf should **not** be installed on older LATEX kernels as it is designed to work with the (New) New Font Selection Scheme, as updated around 2020[3]. Use the initial release of romandadf if your installation of LATEX predates those changes.

Installation varies with TEX distribution so you should consult the documentation which came with your system for details. In most cases, you will need to perform three steps:

1. move or copy the package files to appropriate locations on your system;
2. refresh the TEX database;
3. incorporate the included map file fragments for the different engines your distribution supports.

The following instructions assume you are using TEX Live[4]. They should not be too difficult to adapt if you are using a different distribution.

A.1 Install the files

The files should be installed in one of two locations: *either* the local system-wide TEX tree *or* your personal tree. If the package is installed system-wide, all users will have access to it. On the other hand, you may need privileges you do not have to do this in which case you must use your personal tree.

There are serious disadvantages to installing the package into your personal tree. In particular, these pertain to use of `updmap –user` rather than `updmap –sys`. If you are not aware of these disadvantages, please ensure you are fully cognisant of them before proceeding[5]. Merely removing the package from your personal tree at a later point will *not* undo the effects.

[3]The package should™ work fine on older kernels, but the new version is bound to have some bugs and there is no reason to use it on these systems. The sole purpose of the update is to accommodate the breaking changes made to font selection. If you don't have those changes installed locally, nothing should be broken and the newer version of romandadf offers no advantage at all.

[4]This includes MacTEX for OS X users.

[5]See, for example, Why shouldn't I use getnonfreefonts to install additional fonts? Why shouldn't I use updmap when installing or removing fonts?.

For TEX Live, `kpsewhich -var-value TEXMFLOCAL` will return the path to the local tree and `kpsewhich -var-value TEXMFHOME` the path to your personal tree. The package already includes a hierarchy of files to help you install them correctly. Ignoring any symbolic link in the top directory, move or copy the files in `doc`, `fonts` and `tex` into the appropriate locations. If the tree is initially empty, you can simply move or copy the directories in as they are. If the tree already contains other packages, you may need to merge the package hierarchy with the pre-existing one. For example, if you already have a `doc/fonts` directory, move or copy `doc/fonts/romande` into `doc/fonts/`. If you have a `doc` directory but not a `doc/fonts`, move `doc/fonts` into `doc/`.

A.2 Refresh the database

Again, this depends on your distribution. For TEX Live, `mktexlsr <path to directory>` for the directory you used in the first step should do the trick. Note that you *may* be able to skip this step if you install into your personal tree. Whether this is so depends on the details of your set-up. As a test, move to a directory containing none of the package files and try `kpsewhich romande.sty`. If the file is found, you don't need to refresh the database; otherwise use `mktexlsr` and then try again.

A.3 Install the map fragments

For TEX Live, there are at least two ways of doing this. The second method varies according to the version of TEX Live and instructions are provided accordingly. Both methods depend on whether you installed into `TEXMFLOCAL` or `TEXMFHOME`. If you installed system-wide, the choice is relatively straightforward — it obviously makes sense in that case to update the font maps system-wide as well.

If, on the other hand, you installed into your personal tree, the matter is more complex. On the one hand, updating the system-wide maps may create difficulties or confusion for other users because while the map files will list the fonts as available, they will not be able to access them. On the other hand, maintaining personal font map files can produce difficulties and confusions of its own[6]. Whether it is to be preferred or not is a complex issue and depends on the details of your TEX distribution, local configuration and personal preference. The one clear case is that in which you install into your personal tree because you lack the privileges needed to install system-wide. In that case, you have no choice but to maintain personal font map files or forgo the use of all fonts not provided by your administrator. Other cases are thankfully beyond the scope of this document.

[6] See, for example, Why shouldn't I use `getnonfreefonts` to install additional fonts? Why shouldn't I use `updmap` when installing or removing fonts?.

A.3.1 Method 1

If you installed the package system-wide, use the command:

```
updmap-sys --enable Map=yrd.map
```

If you installed the package in your personal tree, you *may* prefer[7]:

```
updmap --enable Map=yrd.map
```

Either way, `updmap` will output a good deal of information after each incantation. This is normal. Just check that it does not end with an error and that it found the new map file.

A.3.2 Method 2: TEX Live 2008 (and probably earlier)

If you installed the package system-wide, use `updmap-sys --edit`.

If you installed into your personal tree, you *may* prefer to use `updmap --edit`[7].

Either way, a configuration file will be opened which you can edit. Move to the end of the file and add the following line:

```
Map yrd.map
```

When you are done, save the file. `updmap` or `updmap-sys` will produce a great deal of output if all is well. Just check that it does not end with an error and that `yrd.map` is found.

A.3.3 Method 2: TEX Live 2009 (and possibly later)

If you installed the package system-wide, edit or or create `TEXMFLOCAL/web2c/updmap-local.cfg` and add the following line to the end of the file:

```
Map yrd.map
```

Save the file and tell `tlmgr` to merge in your addition using the command:

```
tlmgr generate updmap
```

[7] See, for example, Why shouldn't I use getnonfreefonts to install additional fonts? Why shouldn't I use updmap when installing or removing fonts?.

`tlmgr` will then tell you that you need to ensure the changes are propagated correctly by calling `updmap-sys`. This should produce a great deal of output. Check that it finds the new map file and does not end with an error.

If you installed into your personal tree, you *may* prefer to use `updmap --edit` as described above for TEX Live 2008[8].

A.3.4 Method 3: Current/Recent TEX Live

If you installed the package system-wide, tell `\updmap` to enable the map file:

```
updmap --sys --enable Map=yrd.map
```

This should produce a great deal of output. Check that it finds the new map file and does not end with an error.

If you installed into your personal tree, you *could* use `updmap --user` in place of `updmap --sys` as described above for TEX Live 2008, but this is **not** recommended[8].

To test your installation and that the package works on your system, latex this file (`romandadf.tex`). The console output and/or log should tell you whether any fonts were not found. If you are careful not to overwrite it, you may also compare your output with `romandadf.pdf`.

B Implementation

You do not need to read the remainder of this document in order to install or use the fonts.

To understand how the package is implemented, this section should be read in conjunction with `romandeadf-build.pdf`, which includes codes for custom encodings and 'reglyph' commands, which are necessary to ensure access to the full range of glyphs. This file contains source for only the main 'driver' and 'map' files.

Note that creating the font files, as opposed to just the package and documentation files, *requires* `l3build` and a set of custom `lua` scripts available on `codeberg`. More specifically, if you want to build the font definition files (`.fd`) yourself, you *must* use `l3build fnttarg` and this requires files available from the code repository, but not included in this package.

The reason for this is that `fontinst` provides no way[9] to enable variable scaling. While it is entirely possible to scale a font by any factor you please, it is not, as

[8] See, for example, Why shouldn't I use `getnonfreefonts` to install additional fonts? Why shouldn't I use `updmap` when installing or removing fonts?.

[9] Or no way I've yet discovered.

far as I can tell, possible to enable scaling by any factor a user later pleases. In particular, while it is possible to define shapes and families to use a variable factor, it is not possible to write a definition of that factor into the font definition file, which is the way variable scaling is usually configured.

In order to enable this functionality, lua is used to inject the relevant code into the .fd files after fontinst has generated them. If you simply process the relevant TEX files by hand, you will create broken definition files, since the code produced by fontinst assumes the relevant lines have been injected.

```
\NeedsTeXFormat{LaTeX2e}
\RequirePackage{svn-prov}
\ProvidesPackageSVN[romande.sty]{$Id: romandeadf.dtx 10366 2024-09-18 14:25:21Z cfrees $}[v2.1 \rev-
info]
\DefineFileInfoSVN[romande]
\RequirePackage[T1]{fontenc}
\RequirePackage{nfssext-cfr}[2024/01/01]
```

nfssext-cfr provides \ProcessKeyOptions, \IfFormatAtLeastTF on older kernels.

```
\IfFormatAtLeastTF {2020-02-02}{%
```

To get the oldstyle numbers etc. used from TS1, we need to set the subset to 0 or 1. Unfortunately, this means characters missing from the fonts will not use default symbols as fallback, but this seems to be unavoidable.

```
  \DeclareEncodingSubset{TS1}{yrd}{1}%
  \DeclareEncodingSubset{TS1}{yrda}{1}%
  \DeclareEncodingSubset{TS1}{yrdw}{1}%
  \DeclareEncodingSubset{TS1}{yrdaw}{1}%
}{%
  \RequirePackage{textcomp}}
\UndeclareTextCommand{\textperthousand}{T1}
\ExplSyntaxOn
```

The actual sty is ultra simple. Four options of which only three are actually needed. alt sets the 'alternative' encoding as default or not. A single boolean suffices. scale takes a factor by which to scale the fonts. This is empty by default, which is equivalent to 1, but more efficient.

```
\keys_define:nn { romande }
{
  alt .bool_set:N = \g__romande_alt_bool,
  alt .default:n = true,
  alt .initial:n = false,
  scale .tl_set:N = \yrd@scale,
  scale .initial:V = \@empty,
}
```

Note the optional argument is mandatory in case we're on an older kernel.

```
\ProcessKeyOptions[romande]
```

Use a token list initialised with the bare Berry name. could I just use yrddirectly here? does it matter if it is a cs rather than a tl?

```
\tl_new:N \g___romande_rm_tl
\tl_gset:Nn \g___romande_rm_tl {yrd}
```

Add indicator for 'alt' or not.

```
\bool_if:NT \g___romande_alt_bool
{
  \tl_gput_right:Nn \g___romande_rm_tl {a}
}
```

Order is critical as we're matching on family names. Make RomandeADF default roman font, using the assembled name to implement requested options.

```
\renewcommand{\rmdefault}{\g___romande_rm_tl}
\ExplSyntaxOff
%%%%% end romande.sty
```

The remaining files are not used directly, but are required to generate the files which allow TEX and LATEX to use the fonts. The sources use fontinst as explained in the (sparse) comments. While you can install these files into a TEX tree, they are not required for typesetting.

B.1 Driver

The file does all the initial setup of the fonts. It organises the fonts into families, defines shapes and reencodes as required.

```
\input fontinst.sty
\needsfontinstversion{1.926}
```

Substitutions Bold for bold extended

```
\substitutesilent{bx}{b}
\substitutesilent{b}{db}
\substitutesilent{db}{m}
\substitutesilent{scit}{sc}
\substitutesilent{scsl}{scit}
\substitutesilent{si}{scsl}
\substitutesilent{sl}{it}
```

Record transformations for later map file creation

```
\recordtransforms{yrd-rec.tex}
```

Allow fonts to be scaled via variable in fd files Also requires fontinst.lua fnttarg as no means to define variable in fontinst

```
\declaresize{}{<-> \string\yrd@@scale}
```

Transformations : reencode fonts

```
\transformfont{t1-romandeadf-yrdr}{\reencodefont{t1-romandeadf}{\fromafm{yrdr8a}}}
\transformfont{t1-romandeadf-yrdri}{\reencodefont{t1-romandeadf}{\fromafm{yrdri8a}}}
\transformfont{t1-romandeadf-yrdrc}{\reencodefont{t1-romandeadf}{\fromafm{yrdrc8a}}}
\transformfont{t1-romandeadf-yrdd}{\reencodefont{t1-romandeadf}{\fromafm{yrdd8a}}}
\transformfont{t1-romandeadf-yrddi}{\reencodefont{t1-romandeadf}{\fromafm{yrddi8a}}}
\transformfont{t1-romandeadf-yrddc}{\reencodefont{t1-romandeadf}{\fromafm{yrddc8a}}}
\transformfont{t1-romandeadf-yrdriw}{\reencodefont{t1-romandeadf}{\fromafm{yrdriw8a}}}
\transformfont{t1-romandeadf-alt-yrdr}{\reencodefont{t1-romandeadf-alt}{\fromafm{yrdr8a}}}
\transformfont{t1-romandeadf-alt-yrdri}{\reencodefont{t1-romandeadf-alt}{\fromafm{yrdri8a}}}
\transformfont{t1-romandeadf-alt-yrdd}{\reencodefont{t1-romandeadf-alt}{\fromafm{yrdd8a}}}
\transformfont{t1-romandeadf-alt-yrddi}{\reencodefont{t1-romandeadf-alt}{\fromafm{yrddi8a}}}
\transformfont{t1-romandeadf-alt-yrdriw}{\reencodefont{t1-romandeadf-alt}{\fromafm{yrdriw8a}}}
\transformfont{s-yrdr}{\reencodefont{romande-supp}{\fromafm{yrdr8a}}}
\transformfont{s-yrdri}{\reencodefont{romande-supp}{\fromafm{yrdri8a}}}
\transformfont{s-yrdrc}{\reencodefont{romande-supp}{\fromafm{yrdrc8a}}}
\transformfont{s-yrdd}{\reencodefont{romande-supp}{\fromafm{yrdd8a}}}
\transformfont{s-yrddi}{\reencodefont{romande-supp}{\fromafm{yrddi8a}}}
\transformfont{s-yrddc}{\reencodefont{romande-supp}{\fromafm{yrddc8a}}}
\transformfont{s-yrdriw}{\reencodefont{romande-supp}{\fromafm{yrdriw8a}}}
\transformfont{ts1-yrdr}{\reencodefont{ts1-euro}{\fromafm{yrdr8a}}}
\transformfont{ts1-yrdri}{\reencodefont{ts1-euro}{\fromafm{yrdri8a}}}
\transformfont{ts1-yrdd}{\reencodefont{ts1-euro}{\fromafm{yrdd8a}}}
\transformfont{ts1-yrddi}{\reencodefont{ts1-euro}{\fromafm{yrddi8a}}}
\transformfont{ts1-yrdriw}{\reencodefont{ts1-euro}{\fromafm{yrdriw8a}}}
\input reglyph-yrd.tex
```

Installation: creation of virtual fonts

```
\installfonts
 \installfamily{T1}{yrd}{}
\installfont{yrdr8t}{t1-romandeadf-yrdr,newlatin}{t1-romandeadf}{T1}{yrd}{m}{n}{}
\installfont{yrdrc8t}{t1-romandeadf-yrdrc,newlatin}{t1-romandeadf}{T1}{yrd}{m}{sc}{}
\installfont{yrdri8t}{t1-romandeadf-yrdri,newlatin}{t1-romandeadf}{T1}{yrd}{m}{it}{}
 \installfontas{yrdri8t}{T1}{yrd}{m}{sl}{}
\installfont{yrdd8t}{t1-romandeadf-yrdd,newlatin}{t1-romandeadf}{T1}{yrd}{db}{n}{}
\installfont{yrddc8t}{t1-romandeadf-yrddc,newlatin}{t1-romandeadf}{T1}{yrd}{db}{sc}{}
\installfont{yrddi8t}{t1-romandeadf-yrddi,newlatin}{t1-romandeadf}{T1}{yrd}{db}{it}{}
 \installfontas{yrddi8t}{T1}{yrd}{db}{sl}{}
 \installfamily{T1}{yrda}{}
\installfont{yrdra8t}{t1-romandeadf-alt-yrdr,rs-yrdr,newlatin}{t1-romandeadf-alt}{T1}{yrda}{m}{n}{}
 \installfontas{yrdrc8t}{T1}{yrda}{m}{sc}{}
\installfont{yrdrai8t}{t1-romandeadf-alt-yrdri,rs-yrdri,newlatin}{t1-romandeadf-alt}{T1}{yrda}{m}{it}{}
\installfont{yrdda8t}{t1-romandeadf-alt-yrdd,rs-yrdd,newlatin}{t1-romandeadf-alt}{T1}{yrda}{db}{n}{}
```

```
\installfontas{yrddc8t}{T1}{yrda}{db}{sc}{}
\installfont{yrddai8t}{t1-romandeadf-alt-yrddi,rs-yrddi,newlatin}{t1-romandeadf-alt}{T1}{yrda}{db}{it}
\installfamily{T1}{yrdw}{}
\installfont{yrdriw8t}{t1-romandeadf-yrdriw,newlatin}{t1-romandeadf}{T1}{yrdw}{m}{it}{}
\installfontas{yrdriw8t}{T1}{yrdw}{m}{n}{}
\installfontas{yrdriw8t}{T1}{yrdw}{m}{sc}{}
\installfamily{T1}{yrdaw}{}
\installfont{yrdraiw8t}{t1-romandeadf-alt-yrdriw,rs-yrdriw,newlatin}{t1-romandeadf-alt}{T1}{yrdaw}{m}{it}{}
\installfontas{yrdraiw8t}{T1}{yrdaw}{m}{n}{}
\installfontas{yrdraiw8t}{T1}{yrdaw}{m}{sc}{}
\installfamily{TS1}{yrd}{}
\installfont{yrdr8c}{t1-romandeadf-yrdr,ts1-yrdr,t1-romandeadf-yrdr suffix oldstyle,rs-yrdr,textcomp}{ts1-euro}{TS1}{yrd}{m}{n}{}
\installfontas{yrdr8c}{TS1}{yrd}{m}{sc}{}
\installfont{yrdri8c}{t1-romandeadf-yrdri,ts1-yrdri,t1-romandeadf-yrdri suffix oldstyle,rs-yrdri,textcomp}{ts1-euro}{TS1}{yrd}{m}{it}{}
\installfont{yrdd8c}{t1-romandeadf-yrdd,ts1-yrdd,t1-romandeadf-yrdd suffix oldstyle,rs-yrdd,textcomp}{ts1-euro}{TS1}{yrd}{db}{n}{}
\installfontas{yrdd8c}{TS1}{yrd}{db}{sc}{}
\installfont{yrddi8c}{t1-romandeadf-yrddi,ts1-yrddi,t1-romandeadf-yrddi suffix oldstyle,rs-yrddi,textcomp}{ts1-euro}{TS1}{yrd}{db}{it}{}
\installfamily{TS1}{yrda}{}
\installfontas{yrdr8c}{TS1}{yrda}{m}{n}{}
\installfontas{yrdr8c}{TS1}{yrda}{m}{sc}{}
\installfontas{yrdri8c}{TS1}{yrda}{m}{it}{}
\installfontas{yrdd8c}{TS1}{yrda}{db}{n}{}
\installfontas{yrdd8c}{TS1}{yrda}{db}{sc}{}
\installfontas{yrddi8c}{TS1}{yrda}{db}{it}{}
\installfamily{TS1}{yrdw}{}
\installfont{yrdriw8c}{t1-romandeadf-yrdriw,ts1-yrdriw,t1-romandeadf-yrdriw suffix oldstyle,rs-yrdriw,textcomp}{ts1-euro}{TS1}{yrdw}{m}{it}{}
\installfontas{yrdriw8c}{TS1}{yrdw}{m}{n}{}
\installfontas{yrdriw8c}{TS1}{yrdw}{m}{sc}{}
\installfamily{TS1}{yrdaw}{}
\installfontas{yrdriw8c}{TS1}{yrdaw}{m}{it}{}
\installfontas{yrdriw8c}{TS1}{yrdaw}{m}{n}{}
\installfontas{yrdriw8c}{TS1}{yrdaw}{m}{sc}{}
\endinstallfonts
\endrecordtransforms
\bye
```

B.2 Map

This file is compiled to produce the map file fragment `updmap` needs to install the fonts. It uses files recorded during compilation of the driver.

```
\input finstmsc.sty
```

```
\resetstr{PSfontsuffix}{.pfb}
\adddriver{dvips}{yrd.map}
\adddriver{pltotf}{yrd-pltotf.sh}
\input yrd-rec.tex
\donedrivers
\bye
```

The remaining file listings are included in romandeadf-build.pdf in order to keep the length of this file manageable. Brief descriptions of these files are included below for ease of reference.

B.3 Supplementary (raw)

We need an additional 'raw' encoding to pick up characters otherwise missed. Many of these are here just because they are named differently, but this also covers fancy ligatures, alternate styles of digits etc.

Note that etx files may specify raw and/or output encodings. Those which are specific to romande-TC are described below and included in this file. Those which are not are included as separate sources unless provided by fontinst.

- romande-supp.etx

B.4 Reglyph

We need to rename characters whose names don't match our TeX font encodings.

- reglyph-yrd.tex

B.5 Encodings (output)

These files define variant T1 and TS1 font encodings.

- t1-romandeadf-alt.etx
- t1-romandeadf.etx

In addition to these encodings, we use encoding files supplied by fontinst and some custom files not included in this package's dtx as they are not specific to romandeadf. They are, however, part of the package:

- ts1-euro.etx

[and ts1-euro.enc]

The etx files are not used directly by LaTeX or TeX. Where needed, they are processed to produce enc files. In some cases, however, they are not themselves standalone encodings. Instead, they change how some other encoding is interpreted.

B.6 MTX

mtx files are used to build 'fake' glyphs where these are missing from the original fonts. We do not fake small-caps or bold, but only glyphs which can be constructed without altering the original design.

mtx files are also used to adjust the scale of glyphs and to modify kerning pairs, for example.

We do not use any custom mtx files, though we do use mtx files supplied by fontinst.

C Example

This would be better included in the documentation as a standalone .tex file, but I can't figure out how to do that, so I include the listing here and note that the source for the example can be produced by processing the .ins file with target ee.

```
% !TEX TS-program = pdflatex
% !TEX encoding = UTF-8 Unicode
\documentclass[12pt]{article}
\usepackage{csquotes}
\MakeAutoQuote{‘}{’}
\MakeAutoQuote*{“}{”}
\title{\textswash{Romande ADF Sample (yrd, yrda, yrdaw, yrdw)}}
\author{\textswash{Quick, Q*uesting, Contracting Contrasts Desperately Instantiating Effigies}}

\newcommand{\alphaline}{%
 ABCDEFGHIJKLMNOPQRSTUVWXYZ\par abcdefghijklmnopqrs\kern0pt tuvwxyz\par
 00*123456789 Q Q* f\kern0pt f ff f\kern0pt i fi f\kern0pt l fl f\kern0pt f\kern0pt i ffi f\kern0pt f\kern0pt l ffl c\
 <\kern0pt < << >\kern0pt > >> - -\kern0pt - -- -\kern0pt -\kern0pt - ---\par
  \& \texteuro\ \textdollar\ \textsterling\ \textyen\ \textcurrency\ \textflorin\ \textcent\ \textohm\ \textmu\ \textcelsius\ \textnumero\ \textdiv\ \texttimes\ \textpm\ \textregistered\ \textcopyright\ \texttrademark\ \textparagraph\ \textonequarter\ \textonehalf\ \textthreequarters\ \textonesuperior\ \texttwosuperior\ \textthreesuperior\par
 Sphinx of black quartz, judge my vow.\par
 The quick brown fox jumps over the lazy dog.\par
```

```
143  Querulous sponges act last.\par
144  Q*uerulous s*ponges* act las*t.}
145 \newcommand{\abc}{ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz}
146 \newcommand{\digits}{0123456789}
147 \newcommand{\alphatest}{%
148  {\upshape upright shape:\par \alphaline}\smallskip\par
149  {\scshape small caps:\par \alphaline}\smallskip\par
150  {\itshape italics:\par \alphaline}\smallskip\par
151  {\bfseries
152   {\upshape upright shape:\par \alphaline}\smallskip\par
153   {\scshape small caps:\par \alphaline}\smallskip\par
154     {\itshape italics:\par \alphaline}\smallskip\par
155   }%
156  {\swashstyle script/swash:\par \alphaline}\smallskip\par
157 }
158
159 \pdfmapfile{+yrd.map}
160 \usepackage{romande}
161
162 \begin{document}
163 \setlength{\parindent}{0pt}%
164 \maketitle
165
166 \section*{regular modified encoding}
167
168 \alphatest
169
170 \section*{alternative modified encoding}
171
172 \altstyle
173
174 \alphatest
175
176 \end{document}
```

Change History

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.