The `lstfiracode` package

Ruixi Zhang*

2018/12/24 v0.1c

Contents

1 Introduction

The Fira Code[1] family of fonts, created by Nikita Prokopov, is a monospaced typeface with programming ligatures. It is attempting for many people, me included, to use Fira Code for source code listings. However, the `lstlisting` environment from the `listings` package does not support ligatures naively. To produce the desired output, one must specify all necessary ligatures via the `literate` key of `\lstset`, which can be tedious.

The `lstfiracode` package defines a ready-to-use `listings` style, `FiraCodeStyle`, which pre-specifies 125 ligatures (note that Fira Code v1.206 has 130 ligatures in total).

*ruixizhang42@gmail.com.

[1] See https://github.com/tonsky/FiraCode.

You may *append* the remaining 5 ligatures to the `FiraCodeStyle` literate list via a new key `moreliterate`, without unintentionally erasing all existing ligatures via `literate`.

The `lstfiracode` package also provides a package option, `verbatim`, along with three switches `\ActivateVerbatimLigatures`, `\DeactivateVerbatimLigatures` and `\RestoreVerbatimBehavior` to support source code listings using Fira Code in the `verbatim` environment.

This package does *not* provide the Fira Code font files. The newest version of the fonts can be downloaded at https://github.com/tonsky/FiraCode/releases.

2 Usage

To access `FiraCodeStyle`, simply load `lstfiracode` *after* `listings`. Here is how you may setup your document:

```
\documentclass{article}
\usepackage{fontspec}
\setmonofont{FiraCode-Regular.otf}[
  BoldFont=FiraCode-Bold.otf,
  Contextuals=Alternate  % Activate the calt feature
]
\usepackage{xcolor}
\usepackage{listings}
\usepackage[verbatim]{lstfiracode} % Activate ligatures in verbatim
\lstset{
  language=C++,
  style=FiraCodeStyle,   % Use predefined FiraCodeStyle
  basicstyle=\ttfamily,  % Use \ttfamily for source code listings
  commentstyle=\color{orange}
}
\begin{document}
\begin{verbatim}
A<-www>>=B
\end{verbatim}
\begin{lstlisting}
/* A simple C++ program */
int main() {
    cout << "Hello World"; // prints Hello World
    return 0;
}
\end{lstlisting}
\end{document}
```

which produces the following `verbatim` (observe the `<-`, the `www` and the `>>=` ligatures):

```
A<- www >>= B
```

and the following `lstlisting` (observe the `++` and the `<<` ligatures):

```
/* A simple C++ program */
int main() {
    cout << "Hello World"; // prints Hello World
    return 0;
}
```

3 Package features

3.1 Package option and user commands

The `lstfiracode` package provides one package option and three user commands, described below.

You may load the `lstfiracode` package with the option `verbatim`, or equivalently `verbatim=true`. This activates all Fira Code ligatures in the `verbatim` environment.

```
% Activate Fira Code ligatures in verbatim
\usepackage[verbatim]{lstfiracode}
% is the same as
\usepackage[verbatim=true]{lstfiracode}
```

Loading the package without any option (the default), or equivalently with the option `verbatim=false`, *does not alter* how the `verbatim` environment is handled.

```
% Leave verbatim unaltered
\usepackage{lstfiracode}
% is the same as
\usepackage[verbatim=false]{lstfiracode}
```

You may change your mind in the middle of your document, so there are three switches for such purpose:

`\ActivateVerbatimLigatures` Activate all Fira Code ligatures in `verbatim`. This is executed automatically with the package option `verbatim=true`.

`\DeactivateVerbatimLigatures` Suppress *almost all* Fira Code ligatures in `verbatim`. Currently, it cannot break the `#{` and the `|}` ligatures. You should use Fira Mono if you wish to avoid ligatures altogether.

`\RestoreVerbatimBehavior` Restore how `verbatim` is originally handled by LaTeX.

These switches can overwrite each other and they act *locally*. For instance, the following three `verbatim` environments

```
\begingroup
\ActivateVerbatimLigatures
\begin{verbatim}
A<-www>>=B % Fira Code ligatures activated
\end{verbatim}
\DeactivateVerbatimLigatures
\begin{verbatim}
A<-www>>=B % Fira Code ligatures deactivated
\end{verbatim}
\RestoreVerbatimBehavior
\begin{verbatim}
A<-www>>=B % Hmm...
\end{verbatim}
\endgroup
```

produce, respectively,

```
A<-www>>=B % Fira Code ligatures activated

A<-www>>=B % Fira Code ligatures deactivated

A<-www>=>B % Hmm...
```

3.2 The `FiraCodeStyle` listings style

The ligatures of Fira Code are treated as literate programming by the `lstfiracode` package. These ligatures are specified via the `literate` key in defining `FiraCodeStyle`. The definition of `FiraCodeStyle` looks like this:

```
\lstdefinestyle{FiraCodeStyle}{
  basewidth=0.6em,
  literate=
    {www}{{www}}3
    ...  % All other necessary ligatures in Fira Code
}
```

Thus, `FiraCodeStyle` specifies `basewidth` explicitly and lists *almost all* literate replacements. *It does not contain any font changing commands because users may load Fira Code according to their preferences.* In the case that `\ttfamily` corresponds to Fira Code, be sure to specify `basicstyle=\ttfamily` *in addition to* `style=FiraCodeStyle`, i.e.,

```
\usepackage{listings}
\usepackage{lstfiracode}
% Assume that you have set Fira Code via \setmonofont
% Then remember to also specify basicstyle
\lstset{style=FiraCodeStyle,basicstyle=\ttfamily}
```

3.3 The missing ligatures and the new key `moreliterate`

You may notice that some ligatures in Fira Code are still missing. Well, there are 5 such ligatures — strictly speaking, they *are* listed as literate replacements in the definition of `FiraCodeStyle`, but are simply commented out:

The "missing" ligatures
`/*` `*/` `//` `///` `;;`

These particular combinations of characters usually indicate comment mode. If they were to be implemented as literate replacements, they would break how the `listings` package handles comment highlighting.

Nevertheless, you can still *append* these ligatures to the `FiraCodeStyle` literate list. Say, you want to activate the `;;` ligature in your `C++` code. *But you cannot simply write* `\lstset{style=FiraCodeStyle,literate={;;}{{;;}}2}` *because this will erase all predefined ligatures, leaving only the* `;;` *ligature*. Instead, you should use the new key — `moreliterate` — to add more literate replacements:

```
% Let's add more ligatures
\lstset{
  language=C++,
  style=FiraCodeStyle,
  basicstyle=\ttfamily,
  moreliterate=
    {;;}{{;;}}2
    {///}{{///}}3
}
```

4 Troubleshooting

The `lstfiracode` package is maintained at GitHub. Please make each bug report with a *minimal example* at https://github.com/RuixiZhang42/lstfiracode/issues. Pull requests are welcome.

Version history

v0.1c Removed `Ligatures=Common` from `README.md` and `lstfiracode.tex` (see #1). Re-implemented `\DeactivateVerbatimLigatures`. 2018/12/24

v0.1b Updated `FiraCodeStyle` literate list. Added `\RestoreVerbatimBehavior`. Re-implemented `\Activate/\DeactivateVerbatimLigatures`. 2018/12/20

v0.1a Initial release. 2018/12/16