The \texttt{lstfiracode} package

Ruixi Zhang\textsuperscript{*}

2018/12/24 v0.1c

Contents

1 Introduction 1

2 Usage 2

3 Package features 3

3.1 Package option and user commands .......................... 3

3.2 The \texttt{FiraCodeStyle} listings style ........................ 4

3.3 The missing ligatures and the new key \texttt{moreliterate} .... 5

4 Troubleshooting 5

1 Introduction

The Fira Code\textsuperscript{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 \texttt{lstlisting} environment from the \texttt{listings} package does not support ligatures naively. To produce the desired output, one must specify all necessary ligatures via the \texttt{literate} key of \texttt{\lstset}, which can be tedious.

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

\textsuperscript{*}ruixizhang42@gmail.com.

\textsuperscript{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{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 \textasciitilde, the \texttt{www} and the \texttt{>>=} ligatures):

\( A \textasciitilde \texttt{www} \texttt{>>=} B \)

and the following \texttt{lstlisting} (observe the \texttt{++} and the \texttt{<<} ligatures):

```cpp
/* 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 \texttt{lstfiracode} package provides one package option and three user commands, described below.

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

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

Loading the package without any option (the default), or equivalently with the option \texttt{verbatim=false}, \textit{does not alter} how the \texttt{verbatim} environment is handled.

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

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

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

\texttt{\textbackslash 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.
\texttt{\textbackslash RestoreVerbatimBehavior} Restore how \texttt{verbatim} is originally handled by \LaTeX. These switches can overwrite each other and they act \textit{locally}. For instance, the following three \texttt{verbatim} environments

\begin{verbatim}
\begin{group}
\begin{verbatim}
A<-www>>=B \% Fira Code ligatures activated
\end{verbatim}
\end{group}
\end{verbatim}
\begin{verbatim}
\begin{group}
\begin{verbatim}
A<-www>>=B \% Fira Code ligatures deactivated
\end{verbatim}
\end{group}
\end{verbatim}
\begin{verbatim}
\begin{group}
\begin{verbatim}
A<-www>>=B \% Hmm ...
\end{verbatim}
\end{group}
\end{verbatim}
\end{group}

produce, respectively,

\begin{verbatim}
A<www>>=B \% Fira Code ligatures activated
\end{verbatim}
\begin{verbatim}
A<-www>>=B \% Fira Code ligatures deactivated
\end{verbatim}
\begin{verbatim}
A<www>\geq B \% Hmm ...
\end{verbatim}

3.2 The \texttt{FiraCodeStyle} listings style

The ligatures of Fira Code are treated as literate programming by the \texttt{lстfira} package. These ligatures are specified via the \texttt{literate} key in defining \texttt{FiraCodeStyle}. The definition of \texttt{FiraCodeStyle} looks like this:

\begin{verbatim}
\lstdefinestyle{FiraCodeStyle}{
    basewidth=0.6em,
    literate=
        {www}{{www}}3
        \ldots \% All other necessary ligatures in Fira Code
}
\end{verbatim}

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

<table>
<thead>
<tr>
<th>The “missing” ligatures</th>
</tr>
</thead>
<tbody>
<tr>
<td>/* */ // /// ;;</td>
</tr>
</tbody>
</table>

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