SciTE

Introduction

The SciTE editor is now some 15 years old, and still one of the nicest around. This editor is a wrapper around the scintilla editor framework. It is available for free for Windows and Linux, and there is a relatively cheap version for MacOSX.

Here are a few reasons why I prefer using this editor:

- The footprint is small and (at least on Windows and Linux) installation and updating is easy.
- The editor starts up fast, performs well and the fonts have always been rendered very well.
- Configuration is easy and flexible, and changes to configuration files are reflected immediately (no restart required).
- Lots of file formats are supported by syntax highlighting.
- Processing of files is logged in a fast and efficient log pane.
- There is a Lua interface built-in that permits you to write extensions.

Syntax highlighting

One of the first things that I did when I started using SciTE is to (re)write the TEX and MetaPost lexers so that they were more suitable for ConTEXt. For quite some years ConTEXt has shipped with the relevant files for editing and processing TEX.

Recently the external lexer feature has been extended by an Lpeg based lexer. As a consequence, I wrote a couple of lexers that go beyond the older ones. These are tuned for ConTEXt and are shipped with the ConTEXt distribution.

- The TEX lexer can not only distinguish between tex primitives (including conditionals), low level ConTEXt commands, special registers, ConTEXt user interface commands and special symbols, but can also deal with nested Lua and MetaPost code.
- The Lua lexer has a variant that can recognize some of the ConTEXt MkIV features.
- The xml lexer can recognize some errors in the syntax.
- The Pdf lexer can recognize the relevant objects to some extent.
- The MetaPost lexer can distinguish primitives, MetaFun and user defined commands.

The ConTEXt distribution ships with all the files needed for getting this up and running. In addition to initializing lexers, we also tune some of the menus for use with TEX based workflows. The background of the text areas is set to a light shade of gray and the font defaults to Dejavu Mono; which happens to cover lots of Unicode characters.

The TEX, xml and the yet unmentioned text lexers can do realtime spell checking. As with the lexers, spell checking is more advanced in the Lpeg lexers than in the traditional ones: we recognize rightly spelled words, mark unknown words and also mark words that need case checking. The nice thing is that the regular command highlighting works in parallel. This is shown in figure 1. Normally I use the full (high-res and wide) screen which gives enough room for regular documents as well as the (real-time) log pane. Menus and configured tools adapt themselves automatically to the current file type.

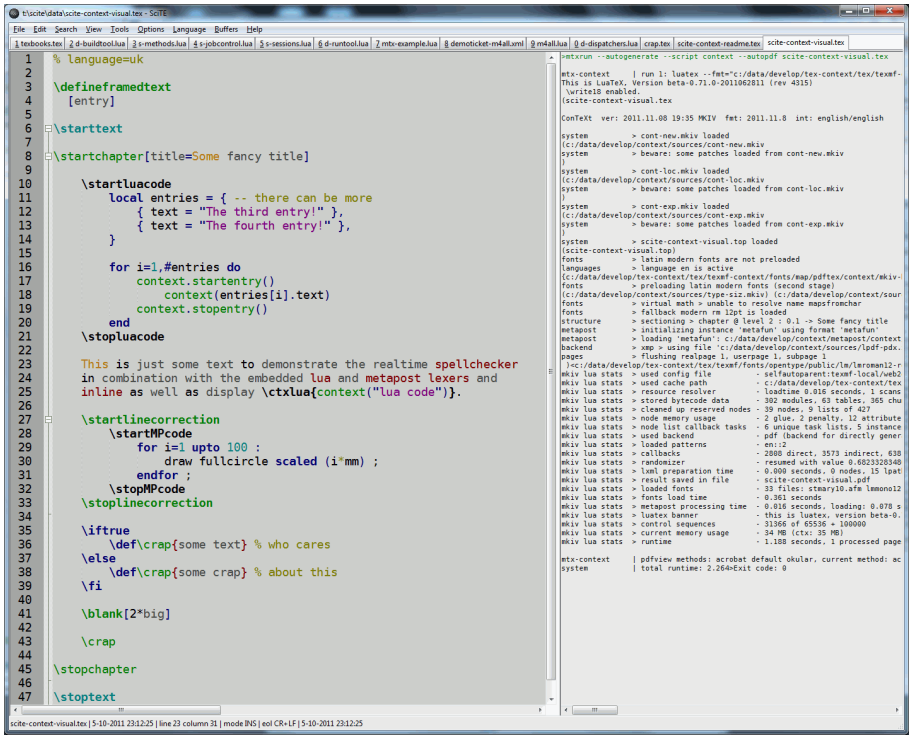

Figure 1. Nested lexers in action.

A side note: currently the MacOSX version does not ship with the library needed for the lpeg lexer. Hopefully this will change one day.

Functionality

Of course, SciTE provides all the regular editing functionality; although in practice one will use only a small subset of features. A real nice feature is the rectangular selection, cut and paste. It's really easy to move columns around. Spacing can be visualized. Structures can be folded.

As an example of a Lua extensions I wrote a word wrapper, simply because I want document sources to look nice as well.

Many files can be open at the same time and are accessed using tabs. The state is remembered and restored at a next startup. Quite handy is the fact that one can collapse all start–ups into one instance.

Word completion is supported (which is quite handy) as is expansion of abbreviation (although I never used that). Also nice is the ability to comment a line or a selected block of text with one key.

Of course, once you are fluent with an editor it stays your favourite no matter what others tell you, but for me it's the only editor that I want to work with.

Side note: a nice alternative is `textadept`, an editor written in Lua using the same scintilla editing component. It shares the Lpeg lexing code but it lacks a realtime log pane and has no tabs.

Processing

Users can easily tweak the `.properties` files to define commands that can be applied to a file. One can add runners to menus and associate them with keys. For TEX compiling, linking et cetera this makes no sense, but checking, processing and

previewing does. For ConTEXt we use `(mtx-)check` for checking, `(mtx-)context` for processing TEX and xml files, `mtxrun` for running Lua, et cetera.

Of course, you can use SciTE for any macro package you like. If you don't use ConTEXt, it provides support for TEX anyway. It's an easy to install editor, so if you're looking for something new it's worth a try.

Hans Hagen