

Editieren mit SciTE

ULRICH HABEL

Es gibt viele verschiedene Editoren. SciTE ist eine interessante Alternative zu den klassischen extbearbeitungswerkzeugen und ganz besonders für Software-Entwickler geeignet – nicht nur, wenn man in Python programmiert.



Die Zahl der Editoren, die um die Gunst der Anwender buhlen, ist fast unüberschaubar. Die Wahl des Editors ist eine sehr persönliche Angelegenheit und führt bisweilen zu beliebten Belustigungen in Foren, wenn sich Emacs- und vim-Anhänger heiße Wortgefechte liefern, als gäbe es nichts wichtigeres und keine anderen Textbearbeitungswerkzeuge. Der hier vorgestellte SciTE ist ein schlanker Editor, der mit ein wenig Aufwand in der Anpassung der Konfigurationsdateien sowohl Systemadministratoren als auch Entwicklern gefallen kann.

Als das Projekt begonnen wurde, war der Editor SciTE gar nicht das eigentliche Entwicklungsziel. Vielmehr plagte sich sein Programmierer Neil Hodgson mit dem Texteditor PythonWin herum, der als Komponente auf dem Richedit-Control basierte, und entwickelte aus dieser Not heraus die Editorkomponente Scintilla. SciTE hatte ursprünglich nur eine Rahmenfunktion um dieses Control herum. Mit Hilfe von SciTE wurde das Steuerelement Scintilla benutzbar und aus dem einstigen Test- und Demoobjekt wurde ein vollwertiger Quelltext-Editor.

Mittlerweile ist der Großteil der Funktionen vollkommen ausgereift. Zu ihnen gehören Codefaltung, Syntax-Hervorhebung und eine erweiterte Konfigurierbarkeit, um nur einige Optionen des Steuerelements zu erwähnen.

Die Installation von SciTE ist denkbar einfach. Viele der großen Distributionen haben SciTE längst in die jeweiligen Paketsysteme aufgenommen, allerdings sind die Versionen oft sehr veraltet. Es ist daher durchaus zu überlegen, ob man nicht die Quelltexte des Programms von [1] herunterlädt und selber kompiliert. Bei Redaktionsschluß dieser freeX-Ausgabe war in den Repositories von Debian Etch beispielweise noch die Version 1.77 zu finden, während aktuell die Version 2.01 heruntergeladen werden kann.

Die Benutzer der Packages Sources von NetBSD sowie der FreeBSD-Ports finden die aktuelle SciTE-Version in den jeweiligen Repositories. Beim Package-Source-System ist der Zweig `pkgsrc-wip` zu nehmen. Die wesentlichen Abhängigkeiten von SciTE sind die `libconv` und das `Gtk 2.8`, alle anderen Bestandteile werden im Quellarchiv mitgeliefert.

Windows-Anwender können gleich unter mehreren Versionen wählen: einem ZIP-Archiv mit allen Dateien, einem Installer sowie einer portablen ausführbaren Datei, die alles benötigte enthält. Vor allem letzteres Standalone-Programm bringt einem Administrator oder Entwickler, der öfters seine Umgebung wechseln muß, unschätzbare Vorteile. Das Binary läßt sich prima auf einen USB-Stick kopieren und ist anschließend überall ohne Installation verfügbar. Selbstverständlich läßt sich SciTE dennoch über eine gesonderte Konfigurationsdatei anpassen.

Der erste Start

Auf der freeX-DVD befinden sich die derzeit verfügbare Extras für SciTE zum Programmieren sowie ein Vorschlag für die initiale Konfiguration, die über den Beitrag hinausgeht. SciTE wurde im Laufe der Zeit immer besser vorkonfiguriert, allerdings lassen vor allem Einstellung wie Schriften und -größen zu wünschen übrig. Die sehr weit gereifte Hilfe ist unter dem Menüpunkt *Hilfe* zu finden, nicht immer führt jedoch der hilfeschuckende Klick zum Erfolg. Wer

SciTE aus der Shell heraus gestartet hat, findet die Antwort auf das Warum schnell: der Aufruf *netscape* wird nicht gefunden. Dieses Problem lässt sich aber über die Konfigurationsdatei schnell lösen.

SciTE unterscheidet verschiedene Konfigurationsdateien, die im Optionsmenü angewählt werden können. Für die Konfiguration der eigenen Einstellung ist das sogenannte »User Options File« zuständig. Es wird immer im Heimatverzeichnis des Benutzers unter dem Namen *SciTEUser.properties* als versteckte Datei abgelegt (unter Windows im Userverzeichnis *%USERPROFILE%*). Um beim Laden der Datei nicht umständlich dorthin navigieren zu müssen, kann die Datei bequem über das Menü *Options* und den Punkt *Open User Options File* geöffnet werden.

Das Durchgehen und Besprechen aller möglichen Konfigurationseinstellungen geht weit über den Rahmen dieser Beschreibung hinaus, es werden hier daher nur die Besonderheiten, die für einen schnellen Start wichtig sind, gezeigt werden. Die Beispiele zum Anpassen des Programms gelten für die Programmiersprache Perl.

Gerade für Anwender, die SciTE auf unterschiedlichen Betriebssystemplattformen nutzen wollen, ist die Unterscheidung wichtig, die SciTE auf der Basis des jeweiligen Interface Toolkits trifft:

```
if PLAT_WIN
    # Anweisungen für Windows
if PLAT_GTK
    # Anweisung für GTK-basierte Systeme
```

Die wichtigste Besonderheit dieser Notation ist die offenstehende *if*-Bedingung. Sie wird automatisch geschlossen, wenn eine Leerzeile den Konfigurationsblock von den nachfolgenden Anweisungen trennt.

Erweiterungen

SciTE beherrscht das Vervollständigen von Quelltext. Diese Codecompletion genannte Funktion besteht einmal aus dem automatischen Ergänzen von Codeelementen und zudem

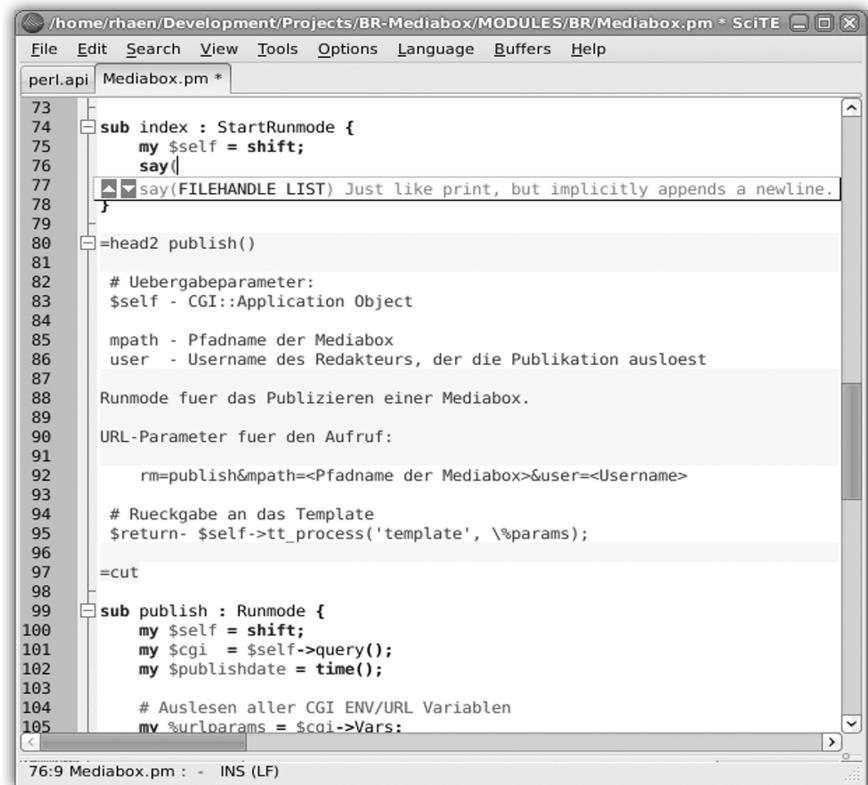


Bild 1: SciTE beim Bearbeiten von Perl-Quelltexten

aus der Anzeige einer Funktionshilfe. Damit diese äußerst praktische Funktionalität möglich ist, müssen vorher von der SciTE-Extras-Seite API-Dateien heruntergeladen werden, die dann in der *SciTEUser.properties* mit der Anweisung *api* aufgenommen werden. Im Beispiel wurde das *perl.api* im Verzeichnis *~/SciTE/* abgelegt. Die Datei gilt für alle Perl-Quelltexte:

```
api.$(file.patterns.perl)=\
    $(SciTEUserHome)/.SciTE/perl.api
```

Die Dateikennungen für die Programmiersprache Perl (*file.pattern.perl*) sind von SciTE mit einem Standardwert vorbelegt. Sie können in der Datei *perl.properties* nachgelesen werden. Die Datei braucht nicht gesucht und manuell geladen zu werden, sondern wird direkt über das Menü *Options | Edit Properties | Open perl.properties* erreicht. In diesen Dateien sollte jedoch nur lesend gearbeitet werden. Voreingestellte Optionen sollten ausschließlich in der *SciTEUser.properties*-Datei überschrieben werden. Die Datei *perl.api* ist leider aber nicht mehr auf dem aktuellen Stand. Sie berücksichtigt zahlreiche Änderungen, die im Laufe der Zeit in Perl Einzug gehalten

haben, nicht mehr. Der Autor dieses Beitrags hat dies zum Anlaß genommen, die bestehende *perl.api* zu erweitern. Auf der [3] können weitere Informationen zu diesem Spezialthema nachgelesen werden, eine Kopie der *perl.api* ist der DVD zu dieser Ausgabe beigelegt.

Wirklich sinnvolles Programmieren ist ohne moderne Werkzeuge nicht mehr vorstellbar. SciTE erkennt an der Dateiendung bereits automatisch den Typ und bietet unter dem Menüeintrag *Tools* diverse vorkonfigurierte Werkzeuge an. Hier findet man eine Syntaxüberprüfung, lint-Kontrolle und das übliche *go*, um das Skript oder das Programm zu starten. Für die Ausgaben der Tools wird ein Output-Fenster geöffnet. Dieses Fenster stellt an sich eine Shell dar, die Kommandozeilenbefehle entgegennimmt. Wem die Art und Weise der vertikalen Fensterdarstellung in SciTE nicht gefällt, kann das natürlich in der Konfigurationsdatei umstellen:

```
split.vertical=0
```

Verwaltungswerkzeuge für Quelltext gehören zum guten Stil, eine Integration in den Editor ist mittlerwei-

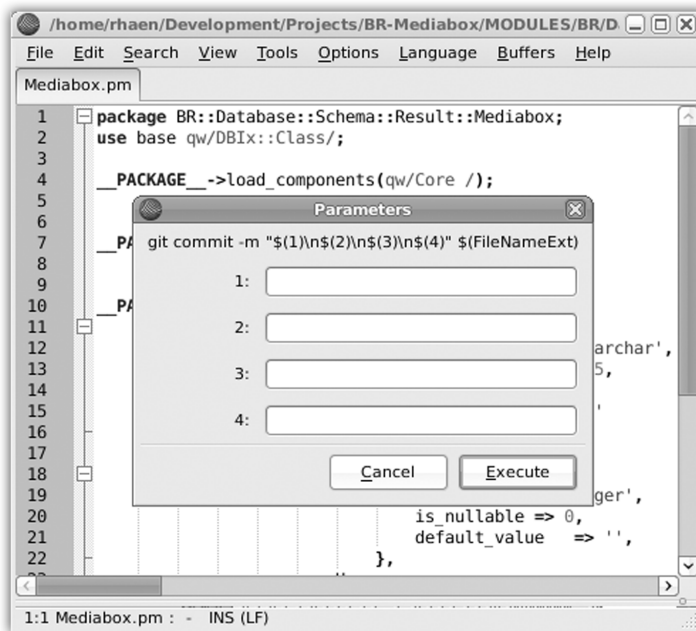


Bild 2: Dialog für Laufzeitparameter

le obligatorisch. SciTE bringt von Hause aus diese Erweiterungen nicht mit, erlaubt aber das einfache Nachrüsten über das *Tools*-Menü. Dort kann man auch eigene Werkzeuge einbringen. Diese Einträge in der Konfiguration des Editors bindet das Versionierungstools *git* ein:

```
# Quelltextverwaltung
command.name.5.*=git commit
command.5.*=\
    *git commit -m "${(1)\n$(2)\n$(3)}" \
    $(FileNameExt)
```

Kommunikation mit der Außenwelt

Besonders zu beachten ist die Notation beim Menüpunkt 5. Durch den Stern vor dem Befehl *git commit* wird ein modaler Dialog mit drei Eingabezeilen angezeigt. Hier kann beispielsweise die Commit-Nachricht für das nachfolgende Kommando auf der Befehlszeile eingegeben werden. Alle anderen Befehle von *git* folgen diesem Beispiel und sind einfach umzusetzen. Durch die beschriebenen Möglichkeiten läßt sich SciTE zu einem funktionsgewaltigen Editor nachrüsten, der kaum noch Wünsche offen lassen dürfte.

Das bisher beschriebene Verfahren für die Eingabe von Parametern kann genauso für den Transfer von Dateien auf einen anderen Server ange-

wendet werden. Die in das Parameterfenster eingegebenen Daten bleiben von Aufruf zu Aufruf erhalten, sie müssen also nicht jedes Mal erneut erfaßt werden. Die Aufnahme von *scp* für das sichere Kopieren des Editorinhalts auf einen anderen Server sieht so aus (wobei in der ersten Zeile des Dialogs der Hostname des entfernten Servers eingetragen werden muß):

```
command.name.1.*=Upload to server
if PLAT_WIN
    command.1.*=pscp.exe $(FileNameExt)\
        "${(1)":
    command.1.subsystem=2
if PLAT_GTK
    command.1.*=scp $(FileNameExt)\
        "${(1)":
```

Durch die Fallunterscheidung bei den Betriebssystemen wird das Werkzeug gleichzeitig für Windows und für unix-basierte Systeme eingerichtet.

Einbinden externer Hilfequellen

Jeder Programmierer und Systemadministrator wird irgendwann an

den Punkt gelangen, am dem der Blick in eine Manpage, die Sprachreferenz oder eine info-Seite erforderlich wird. Zwischen den diversen Fenstern hin und her zu wechseln, ist bei den unix-basierten Systemen zwar problemlos möglich, stört aber den Arbeitsfluß und erfordert weitere Interaktion. SciTE ist in der Lage, beim Aufrufen der Hilfefunktion mit [F1] ein externes Programm zu starten, wobei das geladene Programm abhängig von der Sprache beziehungsweise dem Inhalt des Textfensters sein darf. Bei Perl bietet es sich an, markierte Wörter der Hilfe von *perldoc.org* zu übergeben.

```
command.help.$(file.patterns.perl)=\
cmd /c "start \
http://perldoc.perl.org/search.html?q\
=${CurrentWord}"
```

start ist bei Windows eine Erweiterung der CMD-Befehlszeile und ruft das Programm auf, für das die jeweilige Endung des Parameters registriert ist. In diesem Fall wird der Standardbrowser des Betriebssystems aufgerufen und das aktuell markierte Wort als Suchargument übergeben.

Fazit

Mit SciTE steht, wie diese Ausführungen gezeigt haben, ein schlanker und trotzdem sehr leistungsfähiger Editor zur Verfügung. Die Entwicklung von SciTE ist sehr vielversprechend, der kontinuierliche Releaseprozeß zeigt an, daß die Software am Leben ist.

Die Community ist sehr hilfsbereit und hilft auch unerfahrenen Anwendern gerne weiter. Dank der Möglichkeit, Erweiterungen der Konfiguration in der Programmiersprache LUA zu schreiben, sind den Möglichkeiten fast keine Grenzen mehr gesetzt. ♦

Links:

- [1] SciTE-Homepage: <http://www.scintilla.org>.
- [2] SciTE-Usergroup: <http://groups.google.com/group/scite-interest>.
- [3] perl.api: <http://www.rabenfedern.de/?p=62>.