

# Globaler Autotext (und mehr)

Systemweit – also in jedem beliebigen Programm – Adressen, Grußformeln, Terminologie oder Script-Code abrufen zu können, ist nicht nur für Textprofis unverzichtbar. Das einschlägige Linux-Tool heißt Autokey.

Von Hermann Apfelböck

Die Textverarbeitung kann Autotext, die Tabellenkalkulation automatische Listen, das Mailprogramm kennt Signaturen, die Bash-Shell Aliases. Im Prinzip kann alle diese Einzelaktionen ein zentrales Werkzeug übernehmen – und das arbeitet dann auch in Programmen, die keinerlei Autotext-Funktionalität anbieten. Das englischsprachige Autokey ist ein bewährtes Tool für diese Aufgabe. Es ist ein typisches grafisches Werkzeug für Desktop-Linux und hat dabei zwei sehr unterschiedliche Gesichter: Die Autotextfunktion („Phrases“) ist unkompliziert und jedem Desktopnutzer nachhaltig zu empfehlen. Die (Python-) Script-Funktionalität eröffnet in Verbindung mit Hotkeys und auslösenden Eingabekürzeln interessante Möglichkeiten, ist aber Script-Erfahrenen vorbehalten, weil es an Dokumentation und Debugoptionen mangelt.

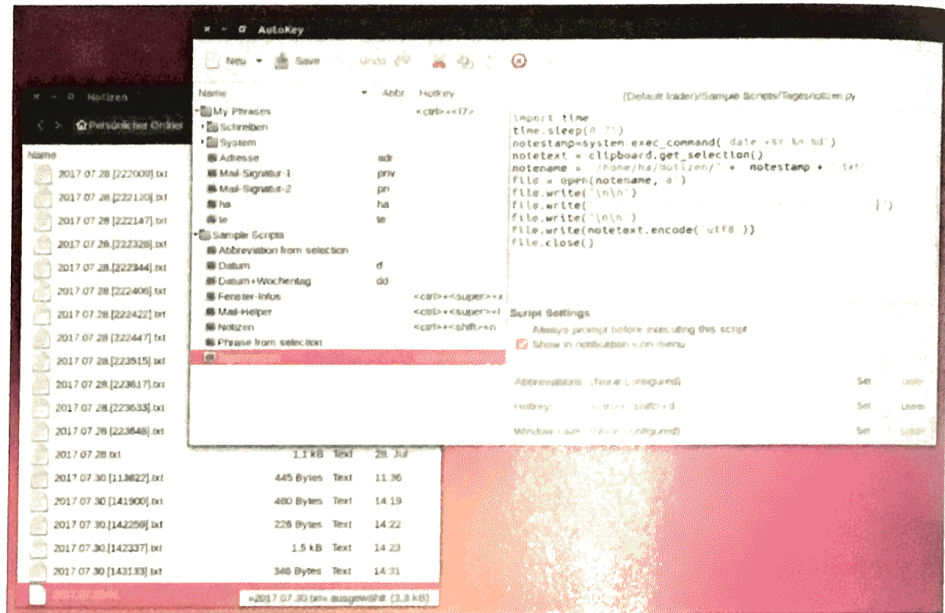
## Autokey installieren und starten

Autokey liegt bei vielen Distributionen in den Standard-Paketquellen und ist daher über den Paketmanager leicht zu installieren, so etwa in Ubuntu oder Linux Mint über das Softwarecenter oder auch auf der Kommandozeile:

```
sudo apt-get install autokey-gtk
```

Die Autokey-Variante für den KDE-Desktop heißt autokey-qt. Falls Ihre Distribution Autokey nicht anbietet, erreichen Sie das Tool auch über das Entwickler-PPA <https://launchpad.net/~cdecker/+archive/ppa>.

Autokey erscheint nach der Installation im Ubuntu-Dash oder im Startmenü. Nach dem Aufruf erscheint der



**Autokey bietet simplen Autotext und komplexe Script-Funktionen: Der hier abgebildete Code für „Tagesnotizen“ wird am Ende des Artikel erklärt.**

Konfigurationsdialog, außerdem nistet sich der Autokey-Indikator dauerhaft im Ubuntu-Hauptpanel mit seinem „A“-Symbol ein. Dieser Indikator bringt Sie jederzeit zum Konfigurationsdialog („Show Main Window“). Unter Ubuntu ist Autokey durch den Panelindikator am besten integriert. In Distributionen wie Linux Mint muss der Konfigurationsdialog über das Startmenü geladen werden. Das ist für die Phase der Makroeinrichtung etwas umständlicher, funktional läuft Autokey aber auch dort einwandfrei.

**Ersteinrichtung:** Sorgen Sie zunächst dafür, dass Autokey automatisch mit Linux startet: Die Option lautet „Edit -> Preferences -> General -> Automatically start AutoKey at login“. Melden Sie sich nach dieser Aktion vom System ab und neu an. Autokey lässt sich zwar sofort konfigurieren und mit neuen Aktionen bestücken, funktioniert aber erst nach der Neuanmeldung korrekt.

**Genereller Tipp:** Wenn Autokey nach diversen Änderungen von Hotkeys und Eingabekürzeln nicht korrekt arbeitet, ist Beenden und Neustart des Tools immer eine gute Antwort. Dann liest das Tool seine Vorgaben unter „~/config/autokey/data“ neu ein, was gelegentliche Konflikte zuverlässig löst. Dieser Hinweis gilt ausschließlich für die Einrichtungsphase.

## Textbausteine definieren

Im Konfigurationsfenster finden Sie unter „My Phrases“ einige vordefinierte Textbausteine. Um eigene Textbausteine anzulegen, verwenden Sie das Menü „New -> Phrase“. Vergeben Sie einen Namen wie etwa „Adresse“ und bestätigen Sie mit „OK“ (dieser Name hat rein organisatorische Funktion und ist nicht mit dem auslösenden Kürzel zu verwechseln, das Sie erst später definieren). Im Editorfenster rechts oben steht „Enter phrase contents“, was Sie

nun durch den tatsächlichen Text ersetzen. Der Text kann aus einem einzigen Wort bestehen oder mehrere Absätze umfassen – wie etwa bei Adressen.

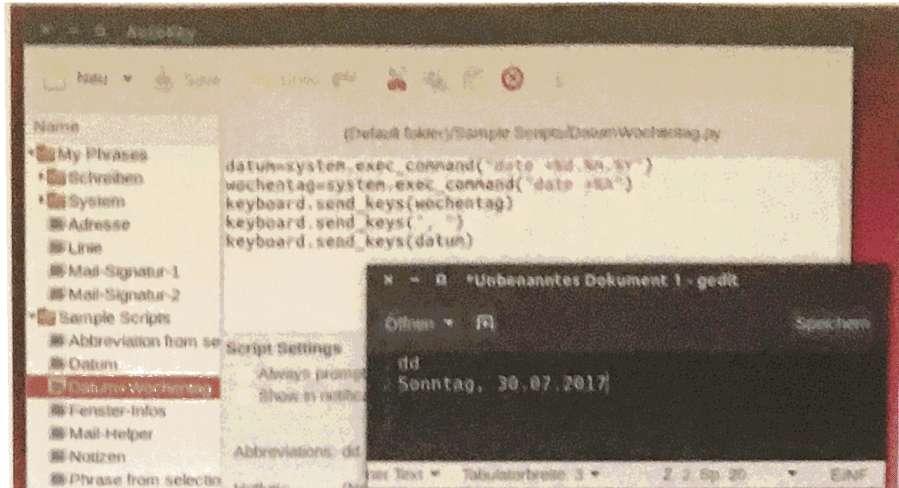
Typischerweise werden Sie einen Textbaustein beim Schreiben durch ein knappes Tastenkürzel auslösen wollen – etwa „adr“ für die Adresse. Dazu klicken Sie neben „Abbreviations“ auf „Set“. Im Unterdialog „Set Abbreviations“ wählen Sie „Hinzufügen“ und geben „adr“ ein. Quittieren Sie das Kürzel immer mit der Eingabetaste – es gibt keine grafische Schaltfläche, um die Kürzeingabe zu bestätigen. Rechts daneben definieren Sie den Auslöser („Trigger on:“). Mit „All non word“ löst jedes Sonderzeichen wie Leerzeichen, Eingabetaste, Tabulator, Punkt oder Bindestrich den Textbaustein aus. Die Eingabe „adr“ und nachfolgendes Leerzeichen würde demnach die komplette Adresse einfügen.

Weitere wichtige Optionen dieses Dialogs sind „Remove typed abbreviation“ und „Omit trigger character“. Aktivieren Sie dies standardmäßig, damit sowohl das Eingabekürzel (etwa „adr“) als auch das Auslöserzeichen (etwa das Leerzeichen) gelöscht werden. „Ignore case...“ löst den Textbaustein auch dann aus, wenn Sie bei der Kürzeingabe die Groß- und Kleinschreibung missachten.

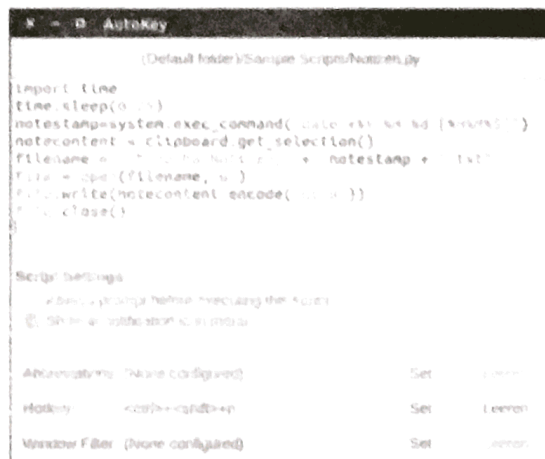
Ist alles definiert, klicken Sie auf „OK“ und im Hauptdialog auf „Save“. Neue Kürzel sind dann sofort aktiv. Wenn Sie das Kürzel in einem beliebigen Programm schreiben und dann das Triggerzeichen folgen lassen, erscheint der komplette Textbaustein.

## Scripts für Texte und weitere Funktionen

Die Script-Fähigkeit von Autokey erlaubt auch dynamische Texteingaben. Script-Automatismen müssen Sie über „Neu -> Script“ als Script definieren – nicht als „Phrase“. Das Miniscript `datum=system.exec_command("date +%d.%m.%Y")` `keyboard.send_keys(datum)` schreibt das aktuelle Datum in der Form „DD.MM.JJJJ“ in das aktuelle



**Dynamischer Textbaustein: Das Kürzel „dd“ gibt das aktuelle Datum im gewünschten Format aus. Das Kürzel wird dabei natürlich wieder gelöscht.**



**Zweiter Notizensammler: Dieser Sammler legt für jeden kopierten Text eine eigene Datei ab. Er ist Alternative oder Ergänzung zum Tagesnotizensammler.**

Programm. Auch dies ist wieder mit einem Kürzel wie etwa „d“ bequem abzurufen. In der Abbildung auf dieser Seite sehen Sie eine andere Datumsvariante mit Wochentag.

Theoretisch eignet sich Autokey mit der Kombination von Hotkey und Mini-Script auch als Programmstarter. Ein einfacher Starter sieht so aus

```
system.exec_command("firefox", True)
```

und muss dann nur noch einen passenden Hotkey erhalten. Da es aber in Desktopdistributionen wie Ubuntu und Mint fast noch einfacher ist, solche Starter in den Systemeinstellungen („Tastatur -> Tastaturkürzel“) anzulegen, gehen wir auf diese Option nicht weiter ein. Die beeindruckenden Möglichkeiten von kleinen Script-Modulen soll ein anderes Beispiel demonstrieren: Hier befördert Autokey beliebige markierte Textschnipsel aus Webseiten,

Mails, Text- oder PDF-Dateien per Hotkey in eine Sammeldatei. Der Script-Code ist im Aufmacherbild dieses Artikels ersichtlich. Das Python-Script „Tagesnotizen“ startet beim Hotkey Strg-Umschalt-D („ctrl-shift-d“ in Autokey. Die Tastenkombination ist natürlich frei wählbar).

Der Beispielcode öffnet oder erstellt dann unter „/home/ha/Notizen“ eine Textdatei mit dem aktuellen Datum und schreibt dorthin den aktuell markierten Text. Ob Sie sich im Browser, im PDF-Viewer, im Terminal oder in einer Textbearbeitung befinden, spielt keine Rolle. Die geringfügig abweichende Variante in der Abbildung auf dieser Seite („Notizen.py“) erstellt für jeden kopierten Text eine eigene Datei mit Zeitstempel.

**Anmerkung:** Alle Script-Beispiele finden Sie auch auf der Heft-DVD unter „/Software/Autokey.txt“.