



TCL/Tk

Eine Einführung

Uwe Berger



Heute schon geTCLt?



Inhalt

- Was ist TCL/Tk?
- TCL Eigenschaften/Besonderheiten
- Die Sprache TCL an Hand von Beispielen
- Grafische Oberflächen mit Tk
- Spracherweiterungen
- Komplexe Anwendungen (Beispiele)
- Informationsquellen



Inhalt

- Was ist TCL/Tk?
- TCL Eigenschaften/Besonderheiten
- Die Sprache TCL an Hand von Beispielen
- Grafische Oberflächen mit Tk
- Spracherweiterungen
- Komplexe Anwendungen (Beispiele)
- Informationsquellen



Was ist TCL?

- TCL - Tool command Language
- eine Scriptsprache wie bash, Perl, php, etc.
- eine Interpretersprache (tclsh, wish)
- auf allen bedeutenden OS-Plattformen verfügbar
- John Ousterhout, ca. 1988
(<http://www.tcl.tk/about/history.html>)
- Open Source



Was ist Tk?

- Tk - Toolkit
- Programmierung von grafischen Benutzeroberflächen (GUI)
- wurde ursprünglich für TCL entwickelt (J. Ousterhout; 1988)
- Plattformübergreifend (soweit grafische Oberflächen verfügbar)
- Anbindung auch für Perl, Python, Ruby und einige weitere vorhanden
- Open Source



Inhalt

- Was ist TCL/Tk?
- **TCL Eigenschaften/Besonderheiten**
- Die Sprache TCL an Hand von Beispielen
- Grafische Oberflächen mit Tk
- Spracherweiterungen
- Komplexe Anwendungen (Beispiele)
- Informationsquellen



TCL Eigenschaften/Besonderheiten

- sehr einfache Syntax-Regeln
- "Alles ist ein String"
- durchgängig Polnische Notation (Befehl Parameter ...)
- alle Datentypen können als String bearbeitet werden (intern natürlich auch Zahlenformate)
- gleiches gilt auch für Programmcode („Alles ist ein String“)
- sehr gute Unterstützung bei der Verarbeitung von Listen und Arrays



TCL Eigenschaften/Besonderheiten

- ereignisgesteuerte Mechanismen für Socket- und Datei-Schnittstellen
- Seriell-, Parallel-, USB- und Netzwerk-Schnittstellen werden als Dateien behandelt
- Zeit- und Benutzerdefinierte Ereignisse
- einfache Ausnahmebehandlung (Fehlerbehandlung)
- einfache Client-/Server-Programmierung
- ... man findet immer wieder neue interessante Dinge ...



Inhalt

- Was ist TCL/Tk?
- TCL Eigenschaften/Besonderheiten
- Die Sprache TCL an Hand von Beispielen
- Grafische Oberflächen mit Tk
- Spracherweiterungen
- Komplexe Anwendungen (Beispiele)
- Informationsquellen



TCL Syntax-Prinzipien

- Kommandowort Parameter1 Parameter2 ...
- Variablen beginnen mit \$ und müssen vor der ersten Verwendung gesetzt sein
- Zeichenketten stehen in doppelten Anführungsstrichen
- Ein Kommando wird von einem Zeilenende oder Semikolon begrenzt
- Backslash (\) am Zeilenende -> Befehl geht auf nächster Zeile weiter



TCL Syntax-Prinzipien

- geschweifte Klammern schützen den Inhalt vor Interpretation (von außen nach innen)
- Code in eckigen Klammern wird zuerst ausgeführt (von innen nach außen)
- Variablen (erstmal) nur lokal gültig und Gültigkeitsbereich muß explizit erweitert werden (global, upvar, uplevel)
- Kommentarzeichen (-befehl): # (am Zeilenanfang oder hinter einem Semikolon)



Starten von TCL-Scripten

- Interaktiv in der TCL-Shell (tclsh, wish) selbst
- Script als Parameter des TCL-Shell:
 - `tclsh tcl_script.tcl`
 - `wish tk_script.tcl`
- Als ausführbares Script mit Angabe des Interpreters im Script-Quelltext selbst:

```
#!/usr/bin/tclsh  
package require Tk  
...
```



TCL in Beispielen

- `bsp_1.tcl`: Variablen, Rechnen, Ausgabe
- `bsp_2.tcl`: Ein-/Ausgabe, Schleifen, if/else
- `bsp_3.tcl`: Filesystem, Listen, Fehlerbehandlung
- `bsp_4.tcl`: Zeitgesteuerte Verarbeitung
- `bsp_5_x.tcl`: Client-/Server-Anwendung, Interpreter
- `bsp_6.tcl`: Aufruf von externen Programmen, Dateiereignis



Inhalt

- Was ist TCL/Tk?
- TCL Eigenschaften/Besonderheiten
- Die Sprache TCL an Hand von Beispielen
- **Grafische Oberflächen mit Tk**
- Spracherweiterungen
- Komplexe Anwendungen (Beispiele)
- Informationsquellen



Grafische Elemente

- „nativ look and feel“
- Tk-Grundelemente (Widgets):
 - Frames, Label
 - Textfeld, Button, Radio-Button, Checkbox, Textbox, Listbox, Menü
 - Canvas, Icons/Bilder
- Fenster, Fensterdekoration
- Systemdialoge, Messageboxen
- Zu jedem Element sind Reaktionen auf Ereignisse (Klick, Selektion etc.) definierbar



Geometriemanager

- bestimmen die Anordnung von Widgets in einem Container (Fenster, Frame) und deren Verhalten bei Veränderung der Containergröße
- ***pack***: Widgets werden nach festlegbaren Regeln automatisch angeordnet
- ***grid***: gezielte Anordnung der Widgets in einem Raster (Zeilen/Spalten)
- ***places***: frei definierbare Anordnung der Widgets in einem Fenster (relat. xy-Koord., Höhe/Breite)



Tk in Beispielen

- `bsp_7.tcl`: „Hello Uwe...“
- `bsp_8.tcl`: Eingabefeld, Radiobutton, MessageBox
- `bsp_9.tcl`: Listbox, Scrollbar, Textbox
- `bsp_10.tcl`: Menües, Systemdialoge
- `bsp_11.tcl`: Zeichnen mit Canvas



Inhalt

- Was ist TCL/Tk?
- TCL Eigenschaften/Besonderheiten
- Die Sprache TCL an Hand von Beispielen
- Grafische Oberflächen mit Tk
- **Spracherweiterungen**
- Komplexe Anwendungen (Beispiele)
- Informationsquellen



Spracherweiterungen

- Packages in TCL/Tk geschrieben
 - z.B. Tcllib, BWidget
- Spracherweiterungen in compilierten Bibliotheken
 - festgelegte Konventionen bei der Funktionsdekleration sind einzuhalten (für C sehr gut dokumentiert)
- „Lib-Wrapper“
 - Einbinden von externen Bibliotheken, die nicht speziell für TCL/Tk geschrieben wurden
 - Sehr instabil, genaue Funktionsweise der externen Bibliothek sollte bekannt sein



Inhalt

- Was ist TCL/Tk?
- TCL Eigenschaften/Besonderheiten
- Die Sprache TCL an Hand von Beispielen
- Grafische Oberflächen mit Tk
- Spracherweiterungen
- Komplexe Anwendungen (Beispiele)
- Informationsquellen



Tipps zur Entwicklung von Tk-GUIs

- wenn möglich nur Geometriemanager pack und grid verwenden: automatische Anordnung der Widgets
- zuerst Oberfläche vollständig konzipieren und als „Programmhülle“ realisieren, dann erst die Funktionalität
- modular arbeiten, viele kleine Prozeduren, die man einzeln testen kann
- möglichst TCL/Tk nur im Standardsprachumfang verwenden: unkomplizierte Weitergabe der Programme
- Fehlerbehandlung nicht vergessen!



Komplexe Anwendungen in TCL/Tk

- `tk_gnupod`: GUI für das Kommandozeilentool `gnupod`
- `tk_id3`: ID3-Tag-Editor
- `tk_proctree`: Programm, zur Anzeige und Manipulation von `/proc`
- `cube_viewer`: Client-/Server-Anwendung um ein ungewöhnliches Display zu simulieren
(<http://www.bralug.de/wiki/3D-LED-Display>)



Inhalt

- Was ist TCL/Tk?
- TCL Eigenschaften/Besonderheiten
- Die Sprache TCL an Hand von Beispielen
- Grafische Oberflächen mit Tk
- Spracherweiterungen
- Komplexe Anwendungen (Beispiele)
- Informationsquellen



Informationsquellen

- „TCL und Tk. Entwicklung grafischer Benutzerschnittstellen für X Window System“; Ousterhout; Addison-Wesley
- „Effektiv TCL/Tk programmieren“; Harrison, McLennan; Addison-Wesley
- <http://www.activestate.com>
- <http://www.tcl.tk>
- <http://wiki.tcl.tk>
- Newsgroup comp.lang.tcl (engl.)