

# Tcl Tools



Carsten Zerbst

# TCL Tools

The number of tools most developers use is quite limited.

Most (Unix-users) seem to work with

- wish or tclsh

# TCL Tools

The number of tools most developers use is quite limited.

Most (Unix-users) seem to work with

- wish or tclsh
- tkcon

# TCL Tools

The number of tools most developers use is quite limited.

Most (Unix-users) seem to work with

- wish or tclsh
- tkcon
- Emacs or Vi (including clones like jed and vim)

# TCL Tools

The number of tools most developers use is quite limited.

Most (Unix-users) seem to work with

- wish or tclsh
- tkcon
- Emacs or Vi (including clones like jed and vim)
- ( debugger )

# ASED

ASED is an editor written in and for Tcl/Tk.

The project was started and is maintained by Andreas Sievers.

The projects homepage is

<http://home.t-online.de/home/andreas.sievers>

# ASEDs Features

- syntaxhighlighting
- undo/redo
- indent/unindent
- find/replace
- test mode
- you can run your code with different wishes
- fast sourcecode navigation through Code Tree Browser
- supporting namespaces as well as itcl
- some code templates
- simple console included
- graphical “grep”
- popup menus (clicking right mouse button)

The screenshot shows the ASEd IDE interface. On the left is the Code Browser showing a project tree with the following structure:

- /home/cz/Projekte/TclProps/Prop
  - <Top>
  - <Bottom>
  - namespace: PropertyService
    - class: PropertiesIterator
      - constructor
      - method: \_Interface
      - method: destroy
      - method: next\_n
      - method: next\_one
      - method: reset
    - class: PropertyNamesIterato
      - constructor
      - method: \_Interface
      - method: destroy
      - method: next\_n
      - method: next\_one
      - method: reset
    - class: PropertySet
      - class: PropertySetDef
      - class: PropertySetDefFactor
      - class: PropertySetFactory

The main editor displays the code for `PropertyService_impl.tcl` and `cpsSetImpl.tcl`. The visible code is as follows:

```

return #retval
}

public method destroy {} {
    set id [$poa reference_to_id [#[this _this]]]
    #poa deactivate_object $id
    unset ::PropertyService::pni#$id
}

private variable poa

class ::PropertyService::PropertiesIterator {
    inherit PortableServer::ServantBase
    constructor {p} {
        set poa $p
    }

    public method _Interface {} {
        return "IDL:org.org/PropertyService/PropertiesIterator:1,0"
    }

    public method reset {} {
        # get the id
        set id [$poa reference_to_id [#[this _this]]]
        set ::PropertyService::pit#$id.(index) -1
    }

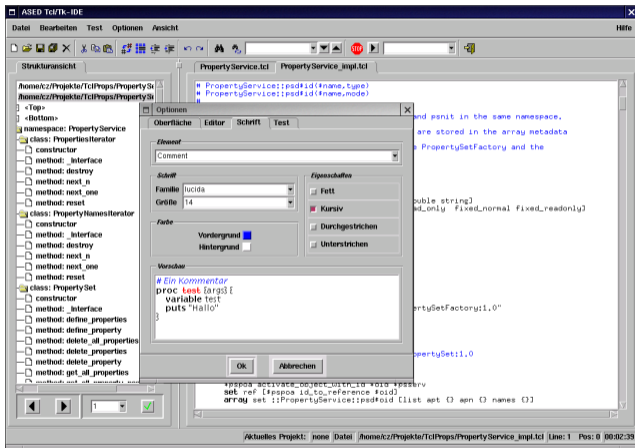
    public method next_one { aproperty } {
        # get the ids
        set id [$poa reference_to_id [#[this _this]]]
        set psd [set ::PropertyService::pit#$id/(psd)]
        set index [incr ::PropertyService::pit#$id/(index)]

        #get the list of names

```

At the bottom of the IDE, the status bar shows: `Current Project: none File /home/cz/Projekte/TclProps/PropertyService_impl.tcl Line: 897 Pos: 0 00:02:12`





# MSGedit

The `msgcat` library is part of the standard distribution.

It provides some functions to manage “message catalogs”. These catalogs consist of arrays, holding the translation to messages. You may choose the keys as you like.

These main functions are

`::msgcat::mclocale` set desired local

`::msgcat::mcset` set translation

`::msgcat::mc` translate

The catalogs are saved in files read by the msgcat library.  
en.msg

```
::msgcat::mcset en "# A comment"  
::msgcat::mcset en "About"  
::msgcat::mcset en "All Files"  
::msgcat::mcset en "Associate a Startfile to the current File"  
::msgcat::mcset en "Attributes"  
::msgcat::mcset en "Auto Completion for ("
```

de.msg

```
::msgcat::mcset de "# A comment" "# Ein Kommentar"  
::msgcat::mcset de "About" "Info"  
::msgcat::mcset de "All Files" "Alle Dateien"  
::msgcat::mcset de "Associate a Startfile to the current File" "Mit einer Startdatei verbinden"  
::msgcat::mcset de "Attributes" "Eigenschaften"  
::msgcat::mcset de "Auto Completion for (" "Ergänzen von ("
```

These files should be named according the ISO 639 country scheme.

It's a tedious task to maintain an correct multiple `.msg` files, even if you just support a small subset like English, French, German ...

Especially synchronizing many languages is a hard task, even with an editor with vertical split like emacs.

To help you editing these files, the editor MSGedit was written. It is available on my homepage

<http://www.tu-harburg.de/~skfcz/tcltk.html>

