# TkDesk User's Guide

Originally written by Christian Bolik, now maintained by J. Chris Coppick, jchris@znet.com Version 1.3, 14 September 1999

TkDesk is a graphical, highly configurable and powerful desktop manager for UNIX and the X Window System. This document is meant to be a comprehensive guide to the functions, services and configuration possibilities offered by TkDesk. Please also take a look at the CHANGES file for latest news. A list of answers to frequently asked questions is also included. The "official unofficial" TkDesk homepage on the Web is at <http://people.mainz.netsurf.de/bolik/tkdesk/>. There is also a mailing list dedicated to TkDesk available at *majordomo@lists.stanford.edu*. To join this list simply send an email with an empty subject line and the body "subscribe tkdesk-list".

## Contents

# 1 Introduction

TkDesk is a graphical desktop manager for UNIX (with a slight emphasis on Linux, but it also runs just as well on AIX, Solaris, HP-UX, SGI Irix and other UNIX flavors) and the X Window System. It offers a very complete set of file operations and services, plus gives the user the ability to configure most aspects of TkDesk in a very powerful way. The reason for this is the use of the Tcl scripting language as the configuration and (for the greatest part of TkDesk) implementation language. This makes TkDesk not just configurable but truly *programmable*. TkDesk has been influenced by various other systems and file managers: NeXT, for laying out the file browser windows, Apple Finder, for the idea of file annotations and, (shock horror), Windows 95, for some other (of course minor and unimportant) inspirations.

This is a brief overview of the most prominent features of TkDesk:

- Arbitrary number of automatically refreshed file browsers and file list windows,
- Configurable file-specific popup-menus,
- Drag and drop,
- Files and directories may also be dropped onto the root window, a.k.a. desktop,
- Configurable application bar, with several displays and cascaded popup menus for each button, files can also be dropped here,
- History of visited directories, opened files, executed commands, and others, which is automatically saved to disk,
- Find files through their annotation, name, contents, size, age, or other criteria,
- A trash can for safe deletion of files and directories,
- Calculation of disk usage for directory hierarchies,
- All file operations (find, copy, disk usage, etc.) are carried out in the background,
- Traversal of directory hierarchies through recursive cascaded menus,
- Bookmarks, create menu entries for often used files/directories,
- Comprehensive hypertextish online help,
- Built-in multi-buffer and undo-capable editor,
- Remote control of XEmacs,
- Close coupling with Netscape Navigator for displaying HTML files or selected URLs,
- Sound support,
- Powerful on-the-fly configuration of nearly all aspect of TkDesk using Tcl/Tk, this also allows the Tcl-literate to extend TkDesk in arbitrary ways,

- Free of charge! But see the file `COPYING`, or menu entry `Help/License` for information on usage and redistribution of TkDesk.

## 1.1 Acknowledgments

TkDesk uses a number of other freely available packages without which TkDesk would not have been possible. I'd like to say many thanks to the following people:

- Chris Sterritt for setting up and managing the TkDesk mailing list (at the old location *majordomo@mrj.com*, this one's disabled since a couple of weeks),
- Alan V. Shackelford for seamlessly taking over the list to his machine and responsibility (the list is now at *majordomo@shaknet.clark.net*),
- Ken Hornstein for his wonderful netscape-remote package,
- Ioi Kim Lan for making an XPM image reader for Tk available,
- George Howlett for his great BLT, of which parts are used by TkDesk,
- Michael McLennan for his massively useful `[incr tcl]`,
- John Ousterhout for Tcl/Tk, which is definitely the best thing since sliced bread,
- Greg Hankins and Matt Welsh for putting together the most wonderful linuxdoc-sgml package,
- and of course, Linus Torvalds whose Linux kind of changed my life, really!

And a very big thank you to the growing TkDesk user community, which provides me with a constant flow of bug reports (getting less now :-)), suggestions for enhancements of TkDesk, and lots of motivation and encouragement.

Special thanks to Chuck Robey for revising a previous version of this guide.

## 1.2 Using TkDesk's Help System

If you check menu entry `Options/Use Netscape for Help`, TkDesk will use that for displaying this User's Guide on-line. Otherwise, to reduce overhead, TkDesk uses its own help system. It features hypertext links, context sensitivity (which is not yet fully utilised by TkDesk) and full text search.

The help window consists of four areas:

1. A **listbox** listing all the section headings. A section can be selected by pressing the left mouse button,
2. the **text display**, which contains the actual help text,
3. a **text entry field** in which a regular expression may be entered (such as `[Ff]eatures`). After hitting `Return`, the whole help text is searched for this expression. Pressing `Return` again continues the search,
4. three **buttons**: "Print" prints the complete help volume, "Back" jumps back after a hypertext link has been followed (see next paragraph), and "Close" to close the help window.

Text that is displayed blue in the help window is a hypertext link. When the left mouse button is clicked over such a link the display will automatically change to the referenced section. You can jump back by pressing the "Back" button described above.

The following keys are bound when the mouse pointer is inside the help window:

`Tab`

Moves to the next section.

`Shift-Tab`

Moves to the previous section.

`Control-Tab`

Moves to the first section.

`Control-Shift-Tab`

Moves to the last section.

`Up, Down`

Scrolls one line up/down.

`Page up, Page down`

Scrolls one page up/down.

`Control-Home`

Jumps to start of help text.

`Control-End`

Jumps to end of help text.

`Meta/Alt-b`

Equivalent to pressing the "Back" button.

`Meta/Alt-c, Escape`

Equivalent to pressing the "Close" button.

## 1.3 Command Line Options

Usually TkDesk is started simply by executing the command "`tkdesk`" from the shell prompt or your X initialisation file. However, you may specify the following options to this command:

**-configdir *dir***

Reads the configuration from directory *dir* instead of `~/.tkdesk`.

**-default**

Reads the default configuration of TkDesk instead of the user's one in `~/.tkdesk`.

**-iconic**

Iconifies all file browser and file list windows created by TkDesk during start-up.

**-layout *file***

Reads the window layout information from file *file* instead of the default `~/.tkdesk/_layout`.

**-startdir *dir***

If this option is given, the first file browser will open with directory *dir*.

**-twm**

Don't use icon windows when file browser or list windows are iconified. Some window managers liek twm cannot handle these properly.

For example, the command "`tkdesk -twm -iconic`" tells Tkdesk to not use icon windows and start with all windows iconified.

# 2 The File Browser Window

The file browser window is the largest window when TkDesk is started for the first time. It contains a menu bar, a configurable button bar (for fast access to the most often used file operations and other functions), an entry field (displaying the current path), a horizontal scrollbar, a certain number of file listboxes (three by default), and a status and information bar.

## 2.1 The Menu Bar

The following sections describe the entries of the individual menus of the file browser windows.

### 2.1.1 TkDesk Menu

The "TkDesk" menu contains the following entries:

**New Browser...**

Asks for a directory for which to open a new file browser window. If the directory is the empty string, the operation is cancelled. By default the checkbox "In Browser" is checked, but if you uncheck it a file list window will be opened instead. The button to the right of the text entry field gives access to a menu containing previously visited directories.

**Clone Window**

Creates a new file browser window with the same directory as the current one (that is, the directory of the file browser from which this menu entry was invoked).

**Display AppBar/Hide AppBar**

Opens the application bar if it had been closed before, or hides (closes) it otherwise. This menu entry's label changes based on whether the AppBar is currently displayed or not.

**Configuration**

This is a submenu from which individual or all configuration files of TkDesk can be opened to modify them.

**Auto Save**

Here you can select which parts of TkDesk are automatically saved periodically and when TkDesk exits.

**Save All Now**

Saves all parts of TkDesk's configuration no matter what the settings in the previous submenu.

**Close Window**

Closes the file browser window.

**Quit**

Quits TkDesk.

### 2.1.2 File Menu

This menu provides all the usual file operations plus entries for finding files. The contained menu entries are:

**Information**

Opens a file information window for each currently selected file. This window, which can also be used to add annotations to files, is described in section 4.1 (File Information).

**New File...**

Asks for the name of a file which will be created in the current directory of the file browser or list. If "Open in Editor" is checked the new file will automatically be opened in the configured editor.

**New Directory...**

Asks for the name of a directory which will be created in the current directory of the file browser or list. If "Open after Creation" is checked the current window will switch to the new directory after it is created.

**Open**

Opens the selected files using their default action, which corresponds to the first action defined in their popup menus as configured in the "Popups" configuration file.

**Print...**

Asks for a command to print the currently selected files. The names of the selected files will be appended to the command. The default command can be defined in the configuration file "System".

**Copy, Move, Link...**

Opens a dialog box for copying, moving, linking, symbolic linking etc. of files. This dialog window is described in more detail in section 4.2 (Copying, Moving and Deleting Files).

**Rename...**

For each selected file, TkDesk asks for a new name. Before actually renaming TkDesk checks for an existing file with the same name.

**Delete...**

Opens a dialog to delete files. The section 4.2 (Copying, Moving and Deleting Files) gives more details.

**Find Files...**

Opens a dialog window which can be used to search for files, using a variety of characteristics. This dialog is described in more detail in section 4.3 (Finding Files).

**Find Annotation...**

Lets you search for files with a certain annotation. More details in section 4.3 (Finding Files).

**Copy To X Selection**

Copies the complete names of all currently selected files (i.e. including their paths) to the X clipboard. They can then be pasted into any other X application using the middle mouse button.

**Copy Names Only**

Same as previous one, but copies only the files' names (i.e. not their paths.)

**History**

Displays a submenu containing recently opened files.

**Close Window**

Closes the window.

### 2.1.3 Directory Menu

The main purpose of this menu is to select directories which are to be opened. It also manages TkDesk's trash can. The menu contains the following entries:

**Open...**

Asks for a directory. A new file list or file browser window will be created, baesed on the setting of the "In Browser" checkbutton.

**New...**

Asks for the name of a directory which will be created in the current directory of the file browser or list.

**Home Directory**

Changes the directory of the window from which this entry is invoked to the user's home directory.

***A number of path entries***

These can be configured in the configuration file `Directories`. See section 9 (Configuration of TkDesk) for details on how to do this. If one of these menu entries is invoked, the path of the file browser will change to that directory. If such an entry is invoked while at the same time pressing the "Control" key, a new file list window will be created (if the option "Always In Browser" is selected a file browser window will be created), displaying the contents of the selected directory. **This feature applies to all menus that contain directory names!**

**Trees**

Contains two cascaded submenus: "Home" and "Root". The contents of these submenus is dynamically generated and corresponds to the directory hierarchy rooted either at the user's home directory or at "/". Selecting an entry from these submenus changes the directory of the window to the selected directory. Pressing `Control` at the same time opens a new window with this directory.

**Open Trash Can**

Opens a file list window displaying the contents of the trash can. When this window is iconified and TkDesk uses icon windows the icon can be used as a Mac-like trash can.

**Empty Trash Can**

Empties the trash can after confirmation from the user. **This erases the files contained in the trash can for good!**

**History**

Displays a submenu containing recently opened directories.

#### 2.1.4 Commands Menu

This menu provides entries to execute commands either once or periodically, edit files, and provides some very basic form of "job control". Its entries are:

**Execute...**

Asks for a command to execute. The button to the right of the command entry contains a history of previously executed commands. Selecting one of these copies its name to the entry widget. Checking the "View Output" button will open an editor window after the command completed displaying its output (stdout and stderr).

**Execute as root...**

The same as the previous entry but TkDesk will ask you for the root password before executing the command. The command will then run under root permission.

**Periodic Execution...**

Opens a window which can be used to execute a command and watch its output periodically. If the "Don't execute" checkbutton is selected, the execution is paused.

**Job Control**

Opens a window which allows to stop, terminate, kill etc. processes which have been started by TkDesk.

**Environment...**

Opens the "Edit Environment" dialog which can be used to display and modify TkDesk's environment variables. Programs and commands started from TkDesk after any modification in this dialog has been made will pick up the modified environment.

**Edit File...**

Asks for the name of a file to edit, and opens it in the editor you selected in the "System" configuration file (defaults to the built-in editor). The "Browse..." button opens a file selector from which a file may be selected.

**New File**

Opens a new blank editor window.

**Edit Selected**

Opens all currently selected files in one new editor window.

**Buffers**

Displays a submenu of all current editor buffers (both files and command output).

***A number of custom command entries***

These can be defined in the configuration file `Commands`. See section 9 (Configuration of TkDesk) for details on how to do this. If one of these menu entries is invoked, the corresponding command will be executed.

**History**

Displays a submenu containing commands recently executed.

### 2.1.5 Bookmarks Menu

This menu lets you create bookmarks for often used files and directories. To do this, select at least one file/directory and invoke the "Add Bookmark" menu entry. The name(s) of the selected files and directories will now appear alphabetically sorted in the "Bookmarks" menu. You can remove any bookmark from the menu by selecting its corresponding entry from the "Remove Bookmark" submenu.

The bookmarks you add will be automatically saved, if the "Bookmarks" entry of the "Auto Save" submenu contained in the "TkDesk" menu is selected (which is the default).

### 2.1.6 Options Menu

The `Options` menu lets you configure many aspects of TkDesk "on the fly". The entries are:

**Long Listing**

Select this to let the details of all files and directories be displayed in the file listboxes. This slightly decreases performance and for that reason is switched off by default.

**Show All Files**

Whether to show files in the file lists whose name start with ".".

**Add Icons**

If selected, small icons will be displayed to the left of the file names in the file listboxes. Turn this off for faster scrolling.

**Folders On Top**

If selected (default) folders will always appear on top of the file lists.

**Append Type Char**

Whether to append a file-type specific character to the file names. This is mainly intended for monochrome displays and is then automatically selected.

**Single Click (Dirs)**

Lets you open directories with a single click of the left mouse button. Individual directories can still be selected by pressing `Control` at the same time.

**Always In Browser**

If this option is selected, the "In Browser" checkbutton of the "Open Directory" dialog will always be selected. `Control-Doubleclick` on a directory will open a file browser instead of a file list window.

**Status in List Windows**

Whether the singly columned file list windows should have a status bar as well.

**Sort by ...**

Sort all file listboxes by one of these criteria: Name, Name (fold, meaning that upper and lower case characters will be "folded", i.e. disregard case when sorting), Size, Date, Extension, or Don't Sort to display directory entries as they are read from disk.

**Strip <your home directory>**

If this is selected, and the current path of the browser is somewhere under your home directory, the leftmost file listbox will contain your home directory rather than the root directory. This speeds up the display of directory hierarchies under your home directory.

**Overwrite Always**

If this option is selected, TkDesk won't ask if the destination file already exists when copying or moving files.

**Really Delete**

Relates to the deletion of files. If this option is selected, the "Delete Files" dialog box will always have the "Delete permanently" checkbutton selected by default.

**Quick Drag'n'Drop**

Normally when you drop files onto a file list, the copy dialog appears. If this option is selected, dropped files will be moved to the destination directory without further questions. If `Control` is pressed during dropping, the files will be copied. If the drop target is the trash can, TkDesk will ask if the files are to be deleted "really".

**Sort History**

This option determines whether the history menus are to be sorted alphabetically or not.

**TkDesk Server**

Start or stop the built-in server to remote execute TkDesk commands. See also section 8 (Using the TkDesk Server).

**Dialogs at Pointer**

If selected, TkDesk will always try to place new windows right under the mouse pointer.

**Autoraise AppBar**

Whether to raise the AppBar above any obscuring window when it is entered by the mouse pointer.

**Use Sound**

If you have sound working with TkDesk on your machine, you can temporarily disable sound by selecting this option. Handy when playing Audio CDROMs, for instance.

**Number Of Listboxes**

This relates to the number of listboxes in the file browser window. Between 1 and 6 can be displayed, though 18 would theoratically not be a problem.

**Balloon Help**

Whether to display a small window at the mouse pointer if it is placed over part of TkDesk, e.g. an AppBar button.

**Use Netscape for Help**

Whether to use Netscape rather than the built-in help viewer for displaying TkDesk online help.

The settings of these options are by default automatically saved to `~/.tkdesk/_options`. This can be disabled by deselecting the "Options" entry of the "TkDesk/Auto Save" submenu.

### 2.1.7 Help Menu

This menu tries to give you some help with using TkDesk. It also contains entries for displaying the list of FAQs, recently made changes to TkDesk, and the license for using and distributing TkDesk:

**User's Guide**

Displays the TkDesk User's Guide either in the built-in help viewer or using Netscape (see previous section).

**Manual Page...**

Asks for a command for which to display the system's manual page in an editor window.

**Getting Started**

Displays the "Getting Started" guide.

**TkDesk FAQ**

Jumps directly to the FAQ section of the TkDesk help volume.

**Changes**

Displays recent changes to TkDesk. Make sure you invoke this entry after upgrading to a newer version of TkDesk.

**License**

Restates that TkDesk is distributed under the GNU Public License (GPL).

**About TkDesk...**

Some info about TkDesk (version, author etc.), plus a link to the TkDesk web page.

## 2.2 The Button Bar

The button bar that's located right underneath the menu bar allows for fast access to any of the commands of TkDesk. It's also possible to configure each button individually to run customized Tcl scripts, that e.g. work on the currently selected files. By default, the buttons in the button bar provide access to the more often used functions of TkDesk (copy, create, delete etc.). When you place the mouse button over any of these buttons, a small help window will appear telling you what this button does (provided the "Balloon Help" option is activated).

The contents of this button bar can be defined via the configuration file `ButtonBar`. See section 9 (Configuration of TkDesk) and the configuration file itself for details on how to do this. Note that by not defining the variable `tkdesk(button_bar)` in that file or by not defining `tkdesk(small_button_bar)` the button for file viewer or list windows may be surpressed respectively.

## 2.3 The Path Entry

This entry field displays the current path of the file browser or file list. You can also type directly into this field to change the display to another directory. There is a sort of "auto-completion" available; if you type only the first part of a directory's name and press `Control-Tab`, its name will be automatically completed as far as there's no ambiguity.

If you click the right mouse button over this field a popup menu appears which lets you select any of the current directory's parent directories and their subdirectories. You can also open files directly from this popup menu.

The button to the right of the entry field contains a menu of the last 30 (default value this may be changed in the "System" configuration file) directories you have visited. If you select one of these, the current path of the browser will change to it. The "Control-trick" to open the selected path in a new list window that's been described in section 2.1.3 (Directories) works here as well!

## 2.4 The File Listboxes

The main part of the file browser and list windows is taken up by the file listboxes. Directories are by default displayed in blue with a bold font, executable files in red and bold, and regular files in black and a medium font. These settings as well as the color, font and icons used for individual file types may be configured via the `FileTags` configuration file. See section 9 (Configuration of TkDesk) for details on how to do this.

Right above each listbox the name of the directory whose contents it shows is displayed, together with the current file mask settings. Clicking on this label reveals a menu with the following entries:

**Refresh**

Refreshes the listbox contents. Although TkDesk refreshes all its file lists every 5 seconds (default), sometimes you may want to have a file listbox instantaneously updated.

**Set Mask...**

Sets a mask for this listbox's display. For example, you can configure the listbox to only display or select files matching the pattern `*.gif`. Multiple masks separated by spaces may be entered to display files matching any of these. A history menu button for reusing masks previously entered is also provided, as well as a checkbutton for inverting the mask's effect (i.e. only files **not** matching the given mask will be displayed).

**No Mask**

Removes a previously activated file mask.

**Disk Usage**

Calculates the directory's disk usage, i.e. the number of kilobytes (default) occupied by that directory and all its subdirectories. A window will appear when the calculation is complete, containing a sorted list of all subdirectories and their individual disk usages. If you double-click on any of its entries, a new file list window will appear displaying the directory's contents. Clicking the right mouse button over any of the directories displays its associated popup menu.

**Free Space**

Displays the space available in the file system of this listbox's directory, using TkDesk's "Periodic Execution" window.

**Execute here...**

Asks for a command to execute in the directory displayed by this listbox. All the usual TkDesk %-sequences to work with the names of selected files may be used, e.g. `%A` to pass all currently selected files to the command.

**Execute as root...**

The same as the previous one, except the command will be run under root permission after asking you for the root password (and you entered the correct one...).

**Long Listing**

Instead of just the file names displays a "long listing" i.e. a listing containing all the details such as size, permission bits, date and time of last modification etc.

**Show All Files**

Whether to display files starting with a dot (".") or not.

**Inverse Order**

Whether to invert the current sorting order.

**Sort by...**

This is identical to the entry of the same name in the menu bar's "Options" menu, but affects this single listbox only.

**Open List Window**

If this listbox is part of a file browser window, open a new list window displaying this listbox's directory.

**Open Browser**

Else, open a browser with this directory.

This menubutton can also be used to drag and drop files, as well as to drag the directory displayed in the associated listbox to another target, e.g. the root window by pressing the middle mouse button over the listbox title menu button.

### 2.4.1 Handling of the Listboxes (Bindings)

The handling of the file browser windows is very similar to that of the NeXT file manager: when you double click on a directory, the listbox to the right of the current one will display this directory's contents. You can open a directory in this listbox, and the listbox to the right of this one will display its contents. This way you can browse through complete directory hierarchies, while having instant access to the contents of the complete directory tree. When the number of opened directories exceeds the number of visible listboxes, you can scroll the listboxes with the horizontal scrollbar which is located right above the file listboxes.

Files are selected in the file listboxes using the left mouse button. A single click selects a single file, deselecting all other files. If the Control key is pressed simultaneously, the old selection will be retained. By dragging the mouse pointer over a file list, while at the same time pressing the left mouse button, a set of files can be selected. Shift-doubleclick selects all files in that file list. Holding down the shift button extends the current selection unto that file (a.k.a. "range-selection").

Note that if the option "Single Click (Dirs)" is set in the "Options" menu a single click on a directory will suffice open it. In this case you need to use Control-Click to select a single directory.

A double click on any listbox item performs its default action. For directories this is to open it, for executables this is to execute it, and for files it invokes the first entry of the corresponding popup menu as defined in the "Popups" configuration file (see below). If the Control key is pressed while double-clicking a directory, a new file list or file browser window (depending on the setting of the "Always In Browser" option) is created displaying the contents of that directory. For files, a dialog box will appear asking for a command to execute on that file.

Files can be dragged by pressing the middle mouse button over any selected file. If no file is selected, the clicked-over file will be selected. Files can be dropped by releasing the mouse button over any other file listbox and the menubutton above them, over windows of the built-in editor, over the application bar (if the corresponding button has been configured to handle dropped files), over iconified file browsers and file lists and in general over any other application supporting the `file` protocol of the BLT package's drag and drop implementation. Valid drop targets can be identified by looking at the window that is displayed while files are dragged: the window's relief appears raised if over a valid target, flat if not.

The right mouse button is used in the file listboxes to access the file-specific popup menu. Every popup menu contains a submenu, labeled with the file's name, which contains entries for the most common file operations. The remaining entries of the menu can be configured via the configuration file `Popups`. See section 9 (Configuration of TkDesk) for details on how to do this.

Since version 1.1 the file listboxes may also be fully controlled using the keyboard:

**Up-/Down Arrow**

Move the selection one file up/down.

**Page Up/Down**

Move the selection one page up/down.

**Home/End**

Move the selection to first/last item in the listbox.

**A-Z**

Select the first file matching the character pressed, cycle through subsequent matches on repeated pressing of the same key.

**Return**

Open selected file.

**Menu (the key left to the right Control key on W95 keyboards)**

Display the currently selected file's popup menu, the menu itself may also be operated using Up/Down, Return, and Esc.

**Backspace**

Change directory to the parent directory.

**F11**

Open the current file listbox's menu.

## 2.5 The Status Bar

The status bar is located at the bottom edge of file browser windows. If "Status in List Windows" is selected in the "Options" menu file list windows will contain a status bar as well.

It displays either

- the current state of TkDesk,
- the amount of free space in the current file system,
- details about the currently selected file, or
- number and total size of currently selected files.

# 3 The File List Window

The file list window is basically a more compact version of the file browser window. It displays only one listbox, which results in the fact that the contents of opened directories are always displayed in the same listbox, and it doesn't have a status bar unless "Status in List Windows" is set in the "Options" menu. Its advantage is that it is more quickly created than a file browser window and occupies less space on your screen. My usual setup is that I have one file browser and lots of file lists opened.

The menu bar of the file list windows is also more compact, but allows access to all menus of the file browser through submenus of the "Others" menu. This menu contains one additional entry to open a file browser window displaying this file list's directory.

# 4 File Operations

## 4.1 File Information

The "File Information" window displays detailed information about a file. It can also be used to change the owner or access-permissions of a file. In addition, an annotation to a file can be added here.

The following information is displayed:

**Path**

The path of the file.

**Size**

The size of the file in bytes.

**Modified**

When the file was last modified. By clicking on the button displaying the date file can be "touched", ie. their modification timestamp will be set to the current time.

**Owner**

Displays the owner of the file. By clicking the button displaying the owner, the owner can be changed to another user.

**Group**

Displays the group ownership of the file. By clicking the button displaying the group's name, the group can be changed.

**Mode**

Here the access permissions are displayed in "ls -l" style. The first three button correspond to the rights of the owner (r: may read, w: may write, x: may execute or open the directory), the second three buttons to the rights of the group, and the last three buttons to every one else's rights. The "x" buttons are special in that they cycle through four settings: `x` for executable, `s` for set user/group id *and* executable, `S` for set id only, and `-` for not executable. Note that using the right mouse button toggles between `-` and `x` only. Clicking the middle mouse button over any of these buttons copies its current setting to the corresponding button in the other two groups. If the settings of any of these buttons is changed, the "Change Mode" button at the bottom edge of the window must be clicked to actually change the file's permissions.

**Links**

Number of (hard) links to this file.

**Type**

Tries to give some information about the file's contents. TkDesk uses the shell command `file` for this.

In the "Annotation" text field you can enter any remarks about the file. This annotation will be saved when the "Close" button is pressed. Files having annotations attached to them will appear underlined in the file listboxes, and their popup menu will contain the first few words of that annotation as an "informative" menu entry.

If the window displays information about a file an additional button labeled "Disk Usage" is provided, which calculates the disk usage of the hierarchy rooted at that directory. The entries of the "Disk Usage" window can be double-clicked to open new file list windows.

### 4.2 Copying, Moving and Deleting Files

These functions can be accessed from the "File" menu, among others. There is one dialog for copying, moving and linking files (plus an arbitrary number of user-defined actions such as diff and patch, as configured in the "Popups" config file), one dialog for renaming files (which does nothing else than moving the file to its new name), and one dialog for deleting files.

The "Rename" dialog is very straight-forward and probably does not need further explanation. The "Copy etc." dialog contains the obvious source and destination entry fields, plus a checkbutton labeled "all selected files" if more that one file was selected when this dialog was opened. If this checkbutton is selected (default) the operation will be applied on all selected files. If it is not checked, each file will be handled individually. The "Skip" button can then be used for skipping individual files.

The "Delete" dialog also contains this checkbutton, plus a checkbutton labeled "Delete permanently". If this checkbutton is selected, the files will *not* be moved to the trash can but will be **really and ultimately deleted!!** The default setting of this checkbutton can be set from the "Options" menu.

### 4.3 Finding Files

The "File" menu contains two entries for finding files: "Find Files..." and "Find Annotation...". Files can be annotated through their "Info" dialog (accessible from the "File" menu or from their popup menu), see section 4.1 (File Information).

"Find Annotation" enables you to look for an annotated file whose annotation matches a certain regular expression (which can be as simple as an ordinary string).

"Find Files" lets you look for files (or rather: let's you instruct TkDesk to look for files) whose names match one or multiple patterns, which are of a certain type (such as directory), which contain a certain string, which are smaller or bigger than a certain number of kilobytes or which are younger/older than a certain date. All fields which are not left blank in this dialog will be combined with a logical AND. This dialog is currently the only one utilizing the balloon help capability of TkDesk, so for now I would like to refer you to this. For instance, if you want to know how to enter the file size you're looking for, place the mouse pointer over the "Size" entry field without moving it for a few seconds.

Both "Find" dialogs display their results in the same file listboxes that are used by the file list and browser windows, so the same bindings described in section 2.4.1 (Handling of the Listboxes (Bindings)) apply here as well!

## 5 Cascading Directory Popup Menus

Ugh, what a title... However, one of the most powerful features of TkDesk are its "cascading directory popup menus." These are menus that start from a given directory, contain submenus for each subdirectories (and possibly menu entries for files), and that dynamically add submenus as you traverse the menu. Hm, kinda hard to explain, but pretty cool. These menus are the ones that are accessible through menu entries "Directory/ Trees/ Home" and "Root", by clicking the right mouse button in the file browser and list window's path entry field, and through the popup menus of directories (submenu "Subdirectories" and "... and Files"). They may also appear in other places, depending on your configuration of TkDesk.

There are two special bindings available in these menus:

**Control-Mousebutton-Release**

On a menu entry associated with a directory this opens a new file list window displaying this directory's contents. On a file it asks for a command to run on this file by displaying the "Open or Execute" dialog.

**Left-and-Right-Mousebutton-Press**

If the left and right mousebuttons are pressed simultaneously over a menu entry, the menu disappear, the popup menu for the associated file or directory is displayed.

Otherwise the directory or file that's selected from the menu is opened using its default command.

# 6 The Application Bar

TkDesk provides you with an application bar for fast access to your favorite applications, commands, directories etc., plus displays for the current date and time, system load, and the status of your mailbox and dialup link. It consists of an arbitrary number of buttons. Each button (also the aforementioned displays) contains a popup menu which can be accessed by pressing the right mouse button over any of them. If you single-click the left mouse button over such a button, the first entry from the corresponding popup menu defining a command to execute will be invoked.

The first button (displaying a comet) could be called TkDesk's "Start Button", although I would rather call it "Take-off Button" ;-). Its popup menu contains entries for accessing TkDesk's most often used functions, such as executing a command or opening a file list or browser window, plus submenus for your bookmarks, the files you most recently opened, and the directories you've last visited. The next entry is a submenu labeled "Application Bar". Here you can configure all aspects of the application bar, especially its **position and orientation**. See my answer to the MFAQ 10.1 (How can I change the position of the application bar?) for more. The last entry labeled "Configuration" contains another submenu which gives you fast access to TkDesk's configuration files.

The second button gives you access to this User's Guide, and lets you view manual pages, also making use of a running TkMan (which is a hypertextified manual pager which is to be highly recommended). This button also allows you to drop executables on it to automatically display their manual page, if they have got one. Note that the default value of the "Manual Page" dialog is the contents of the current X selection, so this can be used for some sort of "context sensitive" help.

All the other buttons can be configured by you! See section 9 (Configuration of TkDesk) for details on how to do this.

# 7 The Built-in Editor

The built-in editor of TkDesk is meant to be a simple ASCII editor for editing files smaller than, say, 500kB. A single editor window can handle an arbitrary number of buffers each of which contains another file. Files can either be loaded through the "File" menu of the editor or by simply dropping one or more files on the text field of the editor window from one of the file listboxes.

An editor window contains the following menus in its menu bar:

**File**

Contains entries to load, save, and print files, and to close the current buffer, current window, or all windows.

**Edit**

The first entry provides access to the editor's "Undo" functionality. Note that the undo buffer may contain a maximum of 500 events. This menu also contains entries for managing TkDesk's own text clipboard. Also provides entries to search for text (regular expressions), replace text, or find the next occurence of the currently selected text. The entry "HyperSearch" is a special form of search: all matching lines are displayed in a file listbox. If one of its entries is clicked on, the editor automatically displays that line at the top of the text field. This can be useful for jumping to headings, function definitions etc. Oh yes, and the regular expressions entered here are also saved with the other histories.

**Options**

The "Auto Indent" option determines whether the cursor is automatically indented after hitting `Return`. If the "Send to Netscape" option is set, the current file will be loaded by (maybe a running) Netscape each time it is saved. Useful when editing HTML files.

**Buffers**

Lists the buffers of the editor window. Buffers can be selected from this menu.

**Configuration**

This menu is only available when editing one of TkDesk's configuration files. It lets you save the file and reload it into TkDesk by selecting the corresponding menu entry or pressing `F5`, or exactly the same plus closing the buffer by invoking the next entry or pressing `F6`. This way configuration files can be edited and reloaded very quickly into TkDesk.

The text area provides all the more common Motif- and Emacs-like key-bindings (including `Control-Space` for selecting an area of the text). Maybe it should be mentioned that `Control-C/X/V` do not work on the X selection but on TkDesk's own clipboard. Use the middle mouse button to paste from the X selection into TkDesk or from TkDesk into another application.

## 7.1 Marks

Marks can be set with `Control-[1-9]`, and be jumped to with `Alt/Meta-[1-9]`. You can always jump back with `Control-0`. They work across buffer and editor window boundaries, and are currently only valid as long as the buffer in which the mark was set is open.

# 8 Using the TkDesk Server

In order to allow remote control of TkDesk, i.e. to allow programs outside of TkDesk to have TkDesk perform certain tasks, TkDesk implements a TCP/IP server that can be used to send Tcl scripts to TkDesk. Whether this server is "up" or not is determined by the option "TkDesk Server" that is to be found in the "Options" menu. Currently, only programs running on the same machine as TkDesk may connect to the server, but multiple TkDesk servers started from different users may be running at the same time.

The server is a much faster way to send command to TkDesk then to use the Tcl/Tk `send` command, as that requires a second Tcl/Tk shell ("wish") to be started to do the send. However, using a TCP/IP server that's basically system-wide accessible to perform arbitrary command under the accout of the user who started TkDesk and thus the server brings with quite a big security risk that TkDesk tries to reduce by keeping the port the TkDesk server uses secret. The port to be used is calculated randomly at server start-up, and is saved into a file that can only be read by the user who started TkDesk. To prevent "guessing" of the port number a generated key is also stored in this file that get passed to the server.

The client performing the communication with the TkDesk server gets installed with TkDesk; its name is `tkdeskclient`. This command expects exactly one argument which will be directly sent to the server and evaluated there as a Tcl script. E.g. you could do a

```
tkdeskclient "dsk_view id"
```

to find out who you are `:-)`.

Along with TkDesk a number of front-end shell scripts for `tkdeskclient` are installed, which comprise of the following:

**cd-tkdesk** ***?path?***

Let's the currently active TkDesk file list or browser window (i.e. the one the mouse pointer was over last) change its directory to *path* if *path* is given, or returns the current directory of the active window.

**ed-tkdesk** ***?+linenum? ?file? ...***

Without argument opens a new editor window, or loads all files given as arguments into the editor (and into the same window if you're using a multi-buffer capable editor, such as the built-in one). If a file is preceded by *+linenum*, i.e. something like *+20*, the built-in editor will position the cursor on the line when displaying the file that's given in the following argument.

**od-tkdesk** ***?dir?***

If no arguments are given opens a file list window for the current directory of the shell/program where the command was issued, or opens a window for the directory specied in *dir*.

**op-tkdesk** ***?file ...?***

For each *file* performs its default action (the one defined first in its corresponding popup menu as defined in "Popups"), or asks for a command to execute if no files are given.

**pop-tkdesk** ***file***

Displays the popup menu for *file*, as defined in the "Popups" config file. The popup menu may also be controlled by the keyboard: Up, Down, Return, and Esc keys do what you would expect.

Note that all of these scripts are merely examples of possible usages of `tkdeskclient`. Use your fantasy to find new and even more exciting applications! `:-)`

# 9 Configuration of TkDesk

Currently, TkDesk can be configured only by editing ASCII files. This is not necessarily a drawback, because this way you can add complex Tcl/Tk procedures to the configuration files. GUI based configuration is planned for one of the next releases.

If you don't know Tcl/Tk: Don't despair! For the biggest part of TkDesk's configuration files it is absolutely not necessary for you to know how to program in Tcl/Tk, since you just have to modify values or extend the examples I and others have provided. And to those who want to exploit all of the power available by using Tcl/Tk as TkDesk's configuration language, please have a look at the answer to FAQ 10.14 (Where can I find out more about Tcl/Tk?).

The configuration files can be accessed from the `TkDesk` menu via the `Configuration` submenu, or from the "Configuration" submenu of the popup menu of the application bar's first button. The built-in editor will then appear with the selected configuration file(s) already loaded. Each configuration file contains pretty detailed comments on what the individual settings are for, and how they can be modified. Well, at least they contain examples, which can guide your modification attempts. Once you have modified the configuration file, you can save it and reload it into TkDesk by making use of the entries of the editor's "Configuration" menu described in section 7 (The Built-in Editor). Use F5 to save the file and reload it into TkDesk, and F6 to do the same but additionally close the configuration file.

As already mentioned, each configuration file contains directions on how to modify its definitions. The general advice is: Simply look at the example definitions, because these make things a great deal clearer than any explanation can.

**Tip:** In case your looking for the definition of a specific part of TkDesk, e.g. for the popup menu of `pdf` files, use the "Find..." entry in the "Configuration" menu. Just enter `pdf` in the "String:" field, hit enter, and double-click the found file.

## 9.1 Configuration Files

All of these are "sourced" (i.e. processed) by TkDesk during start-up. This happens in the following order:

1. System
2. ButtonBar
3. Preferences
4. FileTags
5. Commands
6. Directories
7. Popups
8. AppBar
9. Sounds
10. Local (if existent)

### 9.1.1 AppBar

This configuration file lets you define all aspects of the application bar (apart from its layout, use the entries of the submenu "Application Bar" of the comet-button's popup menu for this). This is what you can define here:

- If the application bar should be managed by the window manager just as an ordinary window. I don't like this, so this is disabled by default `:-)`.
- The maximum number of buttons in a column (vertical layout) or row (horizontal layout). Yes, the application bar can contain several columns or rows!
- The fonts to use for the time and date display.
- Period in seconds after which to update the system load and mailbox displays, where to check for new mail, plus several additional mail related options.
- Icons and commands to use for the dial-up networking button.
- Icons and fonts to use for the trash button, and whether to display its current disk usage in the appbar.
- The actual button definitions, ie. which image to display, and the entries of the (possibly cascaded) popup menu. See below for the syntax used.

### 9.1.2 ButtonBar

This file defines the contents and appearance of the Button Bar, which is displayed underneath the menu bar of TkDesk's file browser (`tkdesk(button_bar)`) or file list windows (tkdesk(small_button_bar). Both button bars are configured independently from one another.

The third list defined here (`tkdesk(dir_button_bar)`) is for directory-specific button bars which will be dynamically displayed and hidden when the respective directory is entered or left. These will be displayed in addition to the standard button bar.

### 9.1.3 Commands

In this file you can define entries which will be added to TkDesk's "Command" menu, for example to uncompress all selected files, manage RPM package files, or do whatever you can possibly imagine.

### 9.1.4 Directories

Here you can define entries and cascaded submenus of directories which will be added to TkDesk's "Directories" menu. More details are contained in the "Directories" file itself.

You can also define a list of directories whose parent directories are not to be displayed in file browser windows, as is usually the case with your home directory. This can speed up display of directories underneath an AFS directory, for instance.

Another list lets you define action that are to be performed when a specific directory is opened. This can be used for example to auto-mount floppies and CDROM's.

### 9.1.5 FileTags

Contains definitions for color and font of specific file types (as distinguished by their extension) as well as standard values for regular files, executables, and directories, that will be used in TkDesk's file lists. Also the icons displayed when the "Add Icons" option is selected are configured here, as well as the icons used for the desk items, i.e. files and directories which have been dropped on the root window.

### 9.1.6 Local

This file does not exist when TkDesk gets installed, but still appears in the "Configuration" menu. This is because if it exists it will be sourced during start-up of TkDesk, so this is the ideal place to put local extensions of TkDesk.

### 9.1.7 Popups

The popup menus that appear when the right mouse button is clicked over a file or directory are defined here. There are individual definitions for the popup menus of directories, executables, and of other files. To find the correct popup menu for a file TkDesk checks the mask of each entry of a popup list one after the other from the beginning of the list to its end, so the masks should become more general towards the end of the list.

An additional popup list (`tkdesk(fileops,popup)`) may be defined here for use in the "Copy, Move, etc." dialog.

#### 9.1.8 Sounds

Here the command to be used for playing sounds can be defined, as well as the sound files which are to be played at certain events. TkDesk comes with a few AU sound files which are located in the `sounds` subdirectory of TkDesk's library directory. You can find out where this is by looking at the fifth line of the `tkdesk` script.

#### 9.1.9 System

All the more "basic" features of TkDesk can be configured here. These are:

- Colours and fonts to be used by TkDesk,
- Default size of file listboxes,
- Definitions of shell commands which are used by TkDesk to copy, move, delete, etc. files. These should be fine as they are in most cases, but you may want to tweak them for your system, so take a look at these definitions.
- The default command for printing,
- Which editor to use, plus settings for the built-in editor,
- Periods for automatic refreshing of the file lists and saving of TkDesk's configuration,
- Maximum number of entries in history menus,
- If TkDesk should ask for confirmation when the user wants to quit TkDesk,
- If TkDesk should allow the menus to be "tearoff-able",
- The paths in which TkDesk will look for images and sounds used in the configuration files,
- Icons to use for file list and browser windows, and for the help window,
- Whether to use "focus follows mouse" default or not,
- Settings for the desk items, e.g. whether to let them be managed by the window manager,
- Commands to execute after start-up and before shutdown (of TkDesk).

### 9.2 The TkDesk API

Many of the values to be specified for TkDesk's configuration files are Tcl scripts. Also, you can define your own Tcl proc's to use in these scripts. To allow better control of TkDesk, TkDesk externalizes a couple of proc's which could be called the "TkDesk API" (although currently they aren't much more then a random set of more or less useful proc's, but anyway).

The proc's I currently think are part of the API are the following:

**dsk_msg_alert** ***msg***

Display *msg* in an alert dialog box.

**dsk_msg_error** ***msg***

Display *msg* in an error dialog box.

**dsk_msg_info *msg***

Display *msg* in an info dialog box.

**dsk_confirm *msg script***

Display *msg* in a dialog box with "OK" and "Cancel" buttons. Evaluate *script* if user pressed "OK".

**dsk_debug *msg***

If TkDesk runs in development mode (e.g. by setting the `-devel` command line option) print *msg* to stderr.

**dsk_ask_dir**

Ask for a directory to open.

**dsk_ask_exec**

Ask for a command to execute or file to open.

**dsk_exec *command***

Execute *command* in the background. When the command completes its exit code will be displayed in the status bar of all file browser windows.

**dsk_exec_as_root *command***

Asks for the root password, and executes *command* in the backgroud under root permission if it was okay.

**dsk_view *command***

Execute *command* in the background and displays its output (stdout and stderr) in a built-in editor window.

**dsk_view_as_root *command***

Same as dsk_view but asks for the root password first.

**dsk_path_exec *path command***

Executes *command* in *path* and in the background.

**dsk_path_view**

Executes *command* in *path* and in the background, and displays its output in an editor window.

**dsk_edit *?+linenum? ?file? ...***

Without arguments asks for a file to open, or opens all *files* in the same editor window, positioning the cursor at line *linenum* if given. If *file* is "New File" a black editor window will be opened.

**dsk_busy**

Displays mouse as "busy" and locks all TkDesk window from receiving mouse events.

**dsk_lazy**

Displays mouse as "lazy", and makes all TkDesk windows responsive again.

**dsk_active *what***

Returns information about the currently active file browser or list window: If *what* is "dir" it returns its directory; if it is "sel" it returns all selected items as a Tcl list; if it is "win" it returns the Tk name of the active window.

**dsk_select X ?names?**

If *names* is not given, copies the full path names of the files selected in the currently active file browser or list window to the X selection. Otherwise just their names are copied.

**dsk_read_string *msg ?script?* ?dontpaste?**

Displays $msg in a dialog box where the user can enter an arbitrary string. If *script* is not given, the string will be returned; if it is given the variable `dsk_read_string` will be set to the user's input and *script* will be evaluated. If "dontpaste" is passed as a third argument to this command the current X selection will not be pasted into the entry field.

**dsk_filesel *message path***

Opens TkDesk file selection dialog using *message* as the label, and presetting the entry field with *path*. *Path* is also used as the filter for the full file selection box.

**dsk_refresh ?file ...?**

For each *file* all windows displaying the corresponding directory is refreshed. *File* may also be a directory.

**dsk_sound *sound***

Plays *sound* if a sound command has been defined and sound is switched on. *Sound* is the second index in the `tkdesk(sound,xxx)` array as defined in the "Sounds" config file ("xxx" in this case).

**dsk_copy ?source ...? ?dest?**

Opens the "Copy, Move, etc." dialog box, filling in "Source" and "Destination" fields if provided.

**dsk_dialup *phonenumber***

Without arguments asks for a phone number, and passes this to the command specified in `tkdesk(appbar,dialup,cmd_up)`, else it does the same without asking.

**dsk_find_files ?options?**

Opens the "Find Files" dialog, presetting its fields from the *options*. The following options are available: -path *path*, -mask *mask*, -string *string*, -regexp *regexp*, -extreg *extreg*, -type "*all|file|dir|sym|socket|pipe*", -owner *owner*, -group *group*, -modified *mod*, -accessed *acc*, -size *size*, -case, -samefs, -followsym. If the option `-doit` is given the search will start as soon as the dialog is displayed, without having to press "Search".

**dsk_mail *file ?string?***

Asks for an email address where to send *file*. If *file* is the empty string, *string* will be sent instead.

**dsk_netscape *what ?url? ?args?***

This proc is used to communicate with a running netscape. If Netscape is not running yet it will be started first. *What* may be "file", "url", or "rcmd" (Communicator only); *url* may be a regular URL, a file name, or, if *what* is "rcmd", it may be "mail", "news" or "edit". *Args* may be "window", or "raise", or both.

**dsk_periodic *command seconds***

Opens the "Periodic Execution" window and executes "command" every "seconds" seconds.

**dsk_open *viewer file***

Opens *file* by performing its default action. *Viewer* should be given as ""; I just didn't manage to get rid of this obsolete argument yet.

**dsk_open_browser** ***dir***

Opens *dir* in a new file browser window.

**dsk_open_dir** ***dir***

Opens *dir* in a new file list window (unless the option "Always in Browser" is set).

**dsk_openall** ***?files?***

Opens all *files* if given, or all currently selected files by performing their respective default action.

## 9.3 Shortcuts

When specifying Tcl scripts in the popup and appbar configuration lists (`tkdesk(popup,...)` and `tkdesk(appbar)`), and in some other places as well (precide, hus?), you can use a number of shortcuts that will be expanded just before evaluation:

**%s**

Will be replaced with the full pathname of the first selected file.

**%f**

Will be replaced with its filename only.

**%b**

Will be replaced with its filename without extension ("basename").

**%d**

Will be replaced with its directory only.

**%A**

Will be replaced with a Tcl list of the full pathnames of all currently selected files, or of the files dropped on an appbar button.

**%a**

Same as %a, but replaces with a list of filenames only.

**%B**

Same as %A but doesn't complain if no files are selected. Instead it will be replaced with the empty string.

**%D**

Will be replaced with the directory of the currently active file list or browser window.

**%x**

Will be replaced with the contents of the X selection.

**%X**

Same as %x but doesn't complain if the selection is empty.

**%S**

For the "Copy, Move, etc." popup menu; will be replaced of what's been entered into the "Source" text entry field.

**%D**

For the same popup menu; will be replaced of what's been entered into the "Destination" text entry field.

### 9.4 Utilities

To make life while configuring TkDesk a little easier as long as there's no GUI configuration available yet, TkDesk provides four little "helper" tools that help with selecting colors, fonts, icons, and sounds, by using graphical mouse-based dialog boxes. These are accessed through the "Configuration" menu, and are basically all handled in the same way.

Each of these dialogs contains three buttons:

**Insert**

Works only correctly if the dialog was invoked from the editor's "Configuration" menu. Inserts the current value at the cursor position in that editor window.

**Select**

Copies the selected value to the X selection, so it can be pasted at arbitrary places using the middle mouse button.

**Close**

Closes the dialog.

## 10 Frequently Asked Questions

### 10.1 How can I change the position of the application bar?

Since version 1.1 the appbar contains a "handle" at the upper or left edge that can be used to drag the appbar around the screen by just pressing the left mouse button over it.

You can also drag the application bar around simply by holding down the Alt- or Meta-key and simultaneously pressing the left mouse button over the application bar. You can also invoke the "Move..." entry from the comet button's popup menu to do the same without having to press Alt/Meta.

Since version 1.0b1 the configuration file "AppBar" sets also a variable named tkdesk(appbar,wm_managed) which can be used to have the apllication bar managed by the window manager, although this is usually not necessary.

### 10.2 Can I have transparent icons?

No, but have a look at the answer to the next question. As far as I know to have transparent icons in X you need to make use of X11's SHAPE extension. Now as raw X programming is something only for the very brave, I didn't look into this any deeper yet. Any takers?

### 10.3 How can I change the background colour of the icons and desk items?

The background colour of icons used when the window manager iconifies a window can be set in the configuration file "System". The variable you are looking for is tkdesk(color,icon_background). By setting this variable to the same colour as your root window you can achieve the effect of transparent icons. You can define the colour either as a normal name (such as "grey", "blue") or in the form #rrggbb.

## 10.4 How can I have a different set of desk items on each virtual screen?

### 10.4.1 FVWM and similar or derived window managers

First, you have to set the variable `tkdesk(desk_items,wm_managed)` in the System config file to 1. Then you have to configure fvwm to not decorate windows of class `dsk_DeskItem`. For instance:

```
Style "dsk_DeskItem"    NoTitle, NoHandles, WindowListSkip, BorderWidth 0
```

### 10.4.2 CDE window manager

As for FVWM, you first have to set the variable `tkdesk(desk_items,wm_managed)` in the System config file to 1. To tell the CDE window manager (dtwm) to not decorate desk items you have to add the following line to either the file Dtwm or .Xdefaults in your home directory, and then restart dtwm:

```
Dtwm*dsk_DeskItem*clientDecoration:  none
```

## 10.5 How do I configure FVWM mini-icons for TkDesk?

For fvwm95's Style command (like fvwm2 from which it comes) you can use a variety of things to identify a tkdesk window - you don't have to use the name only; you can also use the window's class name or resource string. To find out what the window class and resource string are for a certain type of tkdesk window, use the FvwmIdent module (if you use Debian, this will be under the Window Managers->Modules menu; if you're using the default fvwm95rc that comes with the fvwm95 distribution, this will be under the Programs->Modules menu; I don't know about others).

Anyway, FvwmIdent tells me that the window opened by tkDesk's "Open Browser..." menu has a class name of dsk_FileViewer, so I open the FvwmConsole (or FvwmTalk) module, and enter the commands:

```
Style dsk_FileViewer TitleIcon mini-filemgr.xpm
Recapture
```

Then the browser window appears with the appropriate icon. To get the window opened by the "Open Directory..." menu, use: Style dsk_FileList TitleIcon mini-filemgr.xpm

(The "recapture" command is only necessary to apply the command to windows already open; it isn't necessary in your config. file) (Contributed by Daniel Martin, `dtm12@jhunix.hcf.jhu.edu`)

## 10.6 Configuring the appbar or popup menus to execute (export foo=bar; program;) doesn't work.

Yes, this is a bit tricky. What you need to do is the following:

```
dsk_exec sh -c {export foo=bar; program}
```

## 10.7 Composing characters doesn't work in TkDesk's editor.

Currently you have to edit the file cb_tools/bindings.tcl in TkDesk's library directory to make this work. Locate the two lines containing the word "XKB"; all you need to do is to comment out the following lines by prepending a '#'.

## 10.8 TkDesk's layout seems to be screwed. I can't get the appbar displayed anymore.

TkDesk saves all layout information in the file /.tkdesk/_layout, so this is the place to look for bad lines. If in doubt you can safely delete this file and all should be back to normal.

In case of the appbar not showing up you can try the following (sent by Jochem Huhmann, joh@unidui.uni-duisburg.de):

1. Quit tkdesk
2. Open the file ".tkdesk/_layout" within your home directory with your favorite editor
3. Delete the line looking like "Toplevel dsk_appbar 1 66x740+956+0"
4. Write the file back to disk
5. Restart tkdesk. It will place the AppBar according to its defaults now.
6. Place the AppBar as you like (it doesn't like horizontal layout too much)
7. Do a "TkDesk/Save All Now"

## 10.9 On my RedHat 5.x system the appbar clock shows the wrong time.

This is from Bryan Venable, spif@vousi.com:

This may have to do with the location of the zoneinfo directory. For libc5 I believe it's /usr/lib/zoneinfo, whereas for glibc it'd be /usr/share/zoneinfo. Try making a symbolic link from whichever you have to whichever you don't. There is a fix to Red Hat 5.0 which has to do with this, but in that situation the problem is with libc5 programs running on a system "optimized" for glibc.

And Raul Quiroga, quiroga@cartan.math.cinvestav.mx, also has got some advice for this:

Concerning the problem described below I received several suggestions. Some allowed to get the date in the tkdesk right but an incorrect one with the date command. Finally what I did is set the time with timeconfig to Universal with "Hardware clock set to GMT" checked; after a reboot both the appbar and date reported the correct time. Thanks all for your help.

## 10.10 TkDesk complains about "invalid command name wm" and won't start up

There seem to be two solutions to this problem: One is if you're running on a RedHad Linux 5.x system, the libc5 that's packaged and installed in /lib may be too old a version. If `ls /lib/libc.so.5.*` on your system gives something like `/lib/libc.so.5.3.xx` you should upgrade to at least 5.4.20. I think you can get a newer version of libc from sunsite.unc.edu in /pub/Linux/GCC.

The other solution that should always work is to do the following (thanks to Ike Hishikawa, ike@hishikawa.f.uunet.de, for this one):

```
Assuming that the tarball was unpacked under /usr/local,
open /usr/local/bin/tkdesk with your favorite editor.
At the very top of the file it should say:

  #!/bin/sh
  #-*- tcl -*- \
```

```
  PATH=/usr/local/bin:$PATH ;#\
  exec tkdesksh "$0" "$@"

After the 3rd line, insert two lines pointing to the location of
tcl/tk libs,  so that you get:

  #!/bin/sh
  #-*- tcl -*- \
  PATH=/usr/local/bin:$PATH ;#\
  export TCL_LIBRARY=/usr/local/lib/TkDesk/tcl_lib ;#\
  export TK_LIBRARY=/usr/local/lib/TkDesk/tk_lib ;#\
  exec tkdesksh "$0" "$@"

This did the trick for me :)
Hope this helps,
```

## 10.11 I cannot launch other Tcl/Tk applications from within TkDesk

Probably your version of TkDesk sets the environment variables TK_LIBRARY and TCL_LIBRARY, which confuses other Tcl/Tk apps. Unset these variables in the config files before the invocation of the problematic commands, eg. replace xterm with

```
sh -c {unset TK_LIBRARY TCL_LIBRARY; xterm}
```

(contributed by Christoph Dalitz, `dalitz@infotech.de`)

## 10.12 I'd like TkDesk to do this and that. How can I achieve this?

The first place to start are the various configuration files of TkDesk. These can be accessed either by the "TkDesk/Configuration" menu of the file browser windows, or by the "Configuration" submenu of the popup menu of the very first button of the application bar of TkDesk.

Since TkDesk uses Tcl as the language for its configuration, and these configuration files are simply "source"ed, you could add any sort of Tcl proc for instance to the configuration file "System". This proc would then be available in every other configuration file as well. With the set of commands provided by TkDesk, which are listed e.g. in the configuration file "Popups", TkDesk provides a very powerful platform for the user who knows Tcl.

## 10.13 Is there a TkDesk mailing list?

Yes! Thanks to Chris Sterritt and Alan V. Shackelford, there is a mailing list dedicated to TkDesk at shaknet.clark.net. Just send a mail to *majordomo@shaknet.clark.net* with an empty subject line and "subscribe tkdesk" as the message body. Please join, it's worth it! The list's address is *tkdesk@shaknet.clark.net*, by the way.

There is also a searchable archive of the TkDesk mailing list on the web at *http://www.findmail.com/list/tkdesk*. This is probably the best place to start looking when you run into problems with TkDesk.

### 10.14 Where can I find out more about Tcl/Tk?

The official Tcl/Tk homepage is at <http://www.scriptics.com/>. The Tcl/Tk FAQs and lots of user-contributed software can be obtained from <ftp://ftp.neosoft.com/pub/tcl>. There is also a newsgroup dedicated to Tcl/Tk: *comp.lang.tcl*.

## 11 Tips and Tricks

This section currently contains just one tip on how to combine TkDesk and XEmacs. If **you** have any procs or other stuff in your configuration file which you consider could be useful for others as well, or just think it's generally cool, please send me an email, so that I can add it to this section! Thanks!

### 11.1 TkDesk and XEmacs

If you are using XEmacs 19.12 or later you can couple TkDesk and XEmacs quite closely together by adding the following proc into any of your configuration files (I have it in "Popups"):

```
proc xemacs_load {what {where same}} {
    switch $where {
        "same" {
            exec gnudoit -q (find-file \"$what\")
        }
        "other" {
            exec gnudoit -q (find-file-other-window \"$what\")
        }
        "frame" {
            exec gnudoit -q (find-file-other-frame \"$what\")
        }
        "scratch" {
            exec gnudoit -q (switch-to-buffer-other-frame \"*scratch*\")
        }
    }
}
```

And now my generic popup menu for files matching `*` reads:

```
    {{*} {
        {{Edit} {dsk_edit %s}}
        {{XEmacs} {xemacs_load %s}}
        {{Other Window} {xemacs_load %s other}}
        {{Other Frame} {xemacs_load %s frame}}
        -
        {{Print} {dsk_print %s}}
    }}
```

This way you can load files from TkDesk into a running XEmacs! This assumes that you have the command `gnudoit` somewhere in your path, and have started the XEmacs server. This can be done by adding the following line to your `~/.emacs`:

```
(gnuserv-start)
```