Welcome to the ComboBox widget for Motif 1.1 or 1.2 (maybe even 2.0...)!

This is now version 1.32.

THERE ARE A LOT OF NEW RESOURCES AVAILABLE SO PLEASE READ THE

DOCUMENTATION (in ComboBox\_eng.dvi)!!!

If you're in doubt, the primary source of this widget is

ftp.informatik.rwth-aachen.de (137.226.112.172)

The source is in: /pub/packages/ComboBox/ComboBox.tar.gz

This is our campus server in Aachen, Germany. From outside Europe

please also contact:

ftp.x.org (198.112.44.100)

There you'll find the source in:

/contrib/widgets/motif/ComboBox/ComboBox.tar.gz

There are many local ftp servers that mirror the /contrib directory. Thus

with a delay of one or two days you'll also find the source on many other

ftp sites.

PLEASE READ THIS INFORMATION CAREFULLY. IT ALWAYS CONTAINS THE MOST

RECENT INFORMATION.

Whats new? (changes since the last version are marked with an asterisk)

\* Some minor changes concerning the drop down list layout have been made in

order to correct some minor quirks.

\* Updated files for VMS users (hellooooo -- where are you?!!)

- A real man page for use with troff/groff. It's in ComboBox.groff.

- Some applications like GUI development tools or TclMotif were unable to

get notice of these special resources because they were implemented

through the core's setValues method. Thus they don't appeared in the

resource list because there aren't any instance variables connected to

them. These mirror resources instead belong to child widgets. With 1.23

the resource list (which can be queried by XtGetResourceList()) contains

entries for all mirror resources although the initial value setting is

always zero. But this shouldn't matter. You can switch off these entries

by setting -DDONT\_LOOK\_IN\_THE\_MIRROR when compiling ComboBox.c.

- The selection policies XmSINGLE\_SELECT and XmBROWSE\_SELECT are now

supported.

- A new resource called XmNstaticList has been added, which switches the combo

box into a new mode with the drop down list statically displayed.

- Support of XmNautomaticSelection (like the resource known from XmList).

- New callbacks XmNunselectionCallback, XmNdefaultActionCallback,

XmNmodifyVerifyCallback, XmNmotionVerifyCallback and XmNvalueChangedCallback

added (don't know whether I forgot to mention one...).

- New convenience function XmComboBoxClearItemSelection() for deselecting

any selection in the combo box (if XmNselectionPolicy is XmSINGLE\_SELECT).

- New demo app called FontSel. It shows how to realize a full blown

font selection dialog (even for scalable fonts) using combo boxes.

- Some additional casting has been added to the source, so it seems

now really to be ANSI compliant.

- can be compiled with Motif 1.2. (Tested on Sun & HP yet)

- many new resources & a callback that is called whenever the drop down

list gets dropped down or hidden.

- corrected a few minor typos in the header files (mis-spelled function

prototypes) and some minor quirks in the source.

- A FULL-BLOWN ENGLISH DOCUMENTATION (dvi file)

If you can't process dvi files you can download a post script variant

from ftp.informatik.rwth-aachen.de (137.226.112.172) as

/pub/packages/ComboBox/ComboBox\_eng.ps.gz

- The XmNitems and XmNitemCount resources can now be set at any time (not

only when creating the ComboBox widget).

- Corrected some minor memory leaks in the source as well as some typos and

errors in the documentation.

- Drag'n'Drop is now supported with Motif 1.2.

- fvwm's handling of FocusOut events supported.

Well - if you never heard of ComboBoxes you'll have to figure out

yourself what these strange widgets really are. The following text

will only explain how to include it into your own code. (Just as a little

hint: combo boxes are well known from M$ Windoze. A combo box consists

of a text input field with an arrow button besides it. When you click

on the arrow button the combo box pops up a list just below its input

area. You can now browse in it or pop it down by pressing the arrow button

again.)

The source code is now in (wide public) use for nearly a year.

(before this time it was tested for half a year in the Institut fuer

Geometrie und Praktische Mathematik, where I'm working). So you get a well

tested piece of widget magic. Please feel free to report any trouble you

have using the ComboBox widget. But be patient - I'm just a student. In the

meantime I've received a couple of reports and hopefully the widget now can

be used on many well known systems as well as with Motif 1.1 and 1.2. I've

heard of people using the ComboBox widget with Motif 2.0, because they need

the many features... But there seems to be problems with the subclassing

mechanism, so the sanity check CheckComboBox() fails. This check is now

disabled if the source detects a Motif 2.0 release.

BTW - I'm collecting picture postcards...if you have one with a nice picture of

the place you're living in, don't hesitate... 8-)

1. To compile you will need at least Motif :-)

(or :-( depending on your point of view...)

and an \*ANSI\* c compiler (the code is completely ANSI -

no more mess with the old K&R style)!

To compile, try something like

$ cc -DFUNCPROTO -c ComboBox.c

this should do it's work. -DFUNCPROTO is needed because the function

prototypes would be K&R style without it. In addition you often need

to specify to the compiler that the source is ANSI. (-ansi with gcc)

2. Place a

#include "ComboBox.h"

at the top of your source file and create the widget just like any

other ordinary widget. The class is xmComboBoxWidgetClass. There is

no convenience function since it's hardly ever needed with this

widget.

To compile the short demo:

$ cc -DFUNCPROTO ComboBoxDemo.c -o ComboBoxDemo ComboBox.o \

-lXm -lXt -lX11

On some machines you'll need to link additional libs (like network

and socket libs).

Or use the makefile (first edit it accordingly to your needs):

$ make demo

Run the demo with:

$ ./ComboBoxDemo

I don't provide a makefile for every system/compiler combination

because this widget is intended to be placed seamless into the

existing Motif widget set. Thus, compile it and if all is ok, then

copy the includes to the Xm directory and the object file to a

suitable place in your file system -- but don't even think of

/dev/null as a good destination ;-)

To compile the FontSel demo:

Edit the makefile (if you haven't already done so), then do a:

$ make FontSel

And run the Font Selector demo:

$ ./FontSel

If you also have the ButtonFace Library you may want to do a

$ make FontSelX

instead. This will use my pictural push buttons for okay, cancel and help.

3. The ComboBox consists primarily of a XmManager descendant - the

widget you'll get from XtCreateWidget(). This widget contains several

more widgets as its childs: a XmTextField and some other widgets.

You \*must\* not modify any resources of the TextField directly - this

leads sooner or later to trouble. Instead all the necessary instance

variables of the TextField are accessible through the main widget

(the one XtCreateWidget() returned).

There are many (convenience) functions like the ones for XmTextFields

and XmLists to set the TextField's text, add items to the drop down

list and and and... please see ComboBox.h for details (you know:

the documentation is under work...)

4. There is one special resource called "persistentDropDown". It

defaults to "False". Its purpose it not easy to explain since that

has to do with the way your window manager is set up. If you're

using the explicit focus methode (that is: you move the keyboard

focus from application to application by clicking on the top level

window) you're lucky - don't ever think of this resource. For all

those other people (like me): if the window manager is set up to

give the keyboard focus to the top level window the mouse cursor

is over you'll notice one annoying "feature". (It's really \*not\*

a bug ;-) ) If you drop down the ComboBox's list (by clicking on

the arrow right beside the text field) and move the mouse cursor

out of the window the ComboBox resides in the list immediatly

disappears. That happens because the ComboBox looses the keyboard

focus. You can try to bypass that behavior by setting

"persistentDropDown" to "True" if you like. In the application's

resource file a line like

\*YourApp\*ComboBox.persistentDropDown: True

should do the job. But remember: whether this will work depends

on the window manager you're using. I have testet this with

Silicon Graphics' 4dwm (a mwm clone).

5. On some window managers the drop down list gets hidden as soon as

the user moves the mouse cursor into the list. If this happens try

to set the resource "twmHandlingOn" to "True". You can do this

on the command line:

ComboBoxDemo -xrm "\*.twmHandlingOn: True"

This should do it's job with twm as well as fvwm. With olwm you shouldn't

need to set one of the special resources.

6. The combo box has been reported to run successfully on:

Silicon Graphics, Sun, HP, IBM RS6000, PC, even VMS!

Linux, of course...!

Attention VMS folks! You'll find two files called "motif11.opt" or

"motif12.opt" and "vmsmake.com" in this packages. They are provided by

Vincent Li. You'll hopefully know what to do with these two

files. They are provided "as is" with no warranty. Please read the file

"readme.vms".

42. Known bugs: you will need a german dictionary to understand all the

jokes in the source and in the german documentation.

That's it! Enjoy using this ComboBox widget.

(More than 160 downloads yet within the first 3 weeks -- seems that I've

filled in a real gap in Motif. Thanks to all who encouraged me to do

further work on the ComboBox widget -- especially Chris Marotz!)

You may also want to visit my "Motif Programming Cornor" at

file://ftp.igpm.rwth-aachen.de/arc/pub/unix/html/motifcorner.html

with tips & tricks about Motif.

I really don't understand why the folks at OSF didn't implemented

something like the M$ Windoze ComboBox in Motif. Maybe they have

had their moti"f"es... (Rumor has spread it that the mysterious

Motif 2.0 will have one - whow! after many years, the OSF folk got it!)

P.S.: This software is Free Software. Please refer to the file

COPYING, which you should have received together this

file and the source code of the combo box widget.

However if you find this widget to be usefull, you are encouraged

to donate a reasonable amount to a charitable institution --

there are enough people out there who need help! Sometimes I got

the strong impression that programmers tend to get lost in their

own "virtual reality" - sitting in front of their screen, hacking

new algorithms and forget about everything around them. So please

don't forget about people who need help.

(c) 1993, 1994 Harald Albrecht

Institut fuer Geometrie und Praktische Mathematik

Rheinisch-Westfaelische Technische Universitaet Aachen,

D-52056 Aachen, Germany

albrecht@igpm.rwth-aachen.de

Bad Aachen -- Die Stadt der Printe. Warum Aachen ein Kurbad ist?

-- Wieso Kurbad? "Bad" heisst doch nur, dass es hier staendig regnet (wenn

es eben 'mal nicht schneit...)