

KAPL-P-000066
(K-98033)

BUILDING A SIMULATOR CONTROL STATION USING THE TCL/TK LANGUAGE - APPENDIX A
D.R. LaBelle

CONF-980448--

April 1998

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States, nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

KAPL ATOMIC POWER LABORATORY

SCHENECTADY, NEW YORK 12301

Operated for the U. S. Department of Energy
by KAPL, Inc. a Lockheed Martin company

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Building a Simulator Control Station Using the TCL/TK Language
Appendix A

This appendix contains all source code for the Interactive Control Station. The following file contents are included.

ics.tcl
icsserver.tcl
icsgraph.tcl
lib_ics.tcl

```

#wish -f
# Program: ics
# Tcl version: 8.0 (Tcl/Tk/XF)
# Tk version: 8.0
# XF version: 4.0
#
# -----
# Copyright (C) 1996 Dennis R. LaBelle (drlabell@albany.net) All Rights Reserved.
#
# PERMISSION IS GRANTED TO DISTRIBUTE THIS SOFTWARE FREELY, WITH THE
# EXCEPTION THAT ONE MAY NOT CHARGE FOR IT OR INCLUDE IT WITH SOFTWARE
# WHICH IS SOLD. Permission to use, copy, modify, and distribute this software and
# its documentation for educational, research, internal corporate and non-profit
# purposes, without fee, and without a written agreement is hereby granted for all
# cases that do not conflict with the restriction in the first sentence of this
# paragraph, provided that the above copyright notice, this paragraph, and the
# following three paragraphs appear in all copies.
#
# Permission to incorporate this software into commercial products may be obtained
# from the author (e-mail drlabell@albany.net).
#
# IN NO EVENT SHALL THE AUTHOR BE LIABLE TO ANY PARTY FOR DIRECT,
# INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF
# THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE AUTHOR HAS
# BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
# THE AUTHOR SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
# PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS"
# BASIS, AND THE AUTHOR HAS NO OBLIGATION TO PROVIDE MAINTENANCE,
# SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
# -----

```

```

# module inclusion
global env
global xflLoadPath
global xflLoadInfo
set xflMisc(separator) @
set xflLoadInfo 0
if {[info exists env(XF_LOAD_PATH)]} {
  if {[string first $env(XF_LOAD_PATH) @$/lib] == -1} {
    set xflLoadPath $env(XF_LOAD_PATH):@$/lib
  }
  set xflLoadPath @$/lib
}
}
set xflLoadPath @$/lib
}

global argc
global argv
set tmpArgv ""
for {set counter 0} {$counter < $argc} {incr counter 1} {
  case [string tolower [lindex $argv $counter]] in {
    {-xfloadpath} {
      incr counter 1
      set xflLoadPath "[lindex $argv $counter]:$xflLoadPath"
    }
    {-xflstartup} {
      incr counter 1
      source [lindex $argv $counter]
    }
    {-xflbindfile} {
      incr counter 1
      set env(XF_BIND_FILE) "[lindex $argv $counter]"
    }
    {-xflcolorfile} {
      incr counter 1
      set env(XF_COLOR_FILE) "[lindex $argv $counter]"
    }
    {-xflcursorfile} {
      incr counter 1
      set env(XF_CURSOR_FILE) "[lindex $argv $counter]"
    }
  }
}

```

```

{-xfontfile} {
  incr counter 1
  set env(XF_FONT_FILE) "[index $argv $counter]"
}
{-xmodelmono} {
  tk colormodel . monochrome
}
{-xmodelcolor} {
  tk colormodel . color
}
{-xfloding} {
  set xflLoadInfo 1
}
{-xfnloading} {
  set xflLoadInfo 0
}
{default} {
  lappend tmpArgv [lindex $argv $counter]
}
}
}
set argv $tmpArgv
set argc [llength $tmpArgv]
unset counter
unset tmpArgv

```

```

# procedure to show window ShowWindow.alarm
proc ShowWindow.alarm { args } {
# xf ignore me 7

```

```

# build widget .alarm
if {[info procs XFEdit] != ""} {
  catch "XFDestroy .alarm"
} {
  catch "destroy .alarm"
}
toplevel .alarm -relief {raised}

```

```

# Window manager configurations
wm positionfrom .alarm program
wm sizefrom .alarm program
wm maxsize .alarm 1152 864
wm minsize .alarm 479 445
wm protocol .alarm WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .alarm {Alarm Monitor}

```

```

# build widget .alarm.frame1
frame .alarm.frame1 -borderwidth {2} -height {30} -relief {ridge} -width {30}

```

```

# build widget .alarm.frame1.frame
frame .alarm.frame1.frame

```

```

# build widget .alarm.frame1.frame.scrollbar2
scrollbar .alarm.frame1.frame.scrollbar2 -borderwidth {1} -command {.alarm.frame1.frame.listbox1 yview} -highlightthickness {0} -relief {raised}

```

```

# build widget .alarm.frame1.frame.listbox1
listbox .alarm.frame1.frame.listbox1 -background {salmon} -borderwidth {1} -font {-adobe-courier-bold-o-normal-14-140-75-75-m-90-iso8859-1} -yscrollcommand
{.alarm.frame1.frame.scrollbar2 set}

```

```

# build widget .alarm.frame1.frame4
frame .alarm.frame1.frame4 -height {30} -width {30}

```

```

# build widget .alarm.frame1.frame4.label5
label .alarm.frame1.frame4.label5 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal-14-140-75-75-p-82-iso8859-1} -foreground {white}
-relief {raised} -width {2}

```

```

# build widget .alarm.frame1.frame4.label6
label .alarm.frame1.frame4.label6 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {Alarm Time}

```

```

# build widget .alarm.frame1.frame4.label8

```

```

label .alarm.frame1.frame4.label8 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {Current Value}

# build widget .alarm.frame1.frame4.label9
label .alarm.frame1.frame4.label9 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {Low} -width {6}

# build widget .alarm.frame1.frame4.label10
label .alarm.frame1.frame4.label10 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {High} -width {6}

# build widget .alarm.frame1.frame4.label11
label .alarm.frame1.frame4.label11 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {Description}

# build widget .alarm.frame1.message7
message .alarm.frame1.message7 -aspect {1500} -background {salmon} -borderwidth {0} -font {-adobe-helvetica-bold-o-normal--14-140-75-75-p-82-iso8859-1} -justify
{center} -padx {6} -pady {3} -text {A
L
A
R
M
I
N
G}

# build widget .alarm.frame3
frame .alarm.frame3 -height {30} -width {30}

# build widget .alarm.frame3.button14
button .alarm.frame3.button14 -borderwidth {1} -command {DestroyWindow.alarm} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11}
-pady {1} -state {active} -text {Done}

# build widget .alarm.frame3.button15
button .alarm.frame3.button15 -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {1} -text {Clear}

# build widget .alarm.frame3.checkbutton0
checkbutton .alarm.frame3.checkbutton0 -borderwidth {1} -relief {raised} -text {audible} -variable {alambell}

# build widget .alarm.frame12
frame .alarm.frame12 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .alarm.frame12.frame
frame .alarm.frame12.frame

# build widget .alarm.frame12.frame.scrollbar2
scrollbar .alarm.frame12.frame.scrollbar2 -borderwidth {1} -command {.alarm.frame12.frame.listbox1 yview} -highlightthickness {0} -relief {raised}

# build widget .alarm.frame12.frame.listbox1
listbox .alarm.frame12.frame.listbox1 -background {mediumaquamarine} -borderwidth {1} -font {-adobe-courier-bold-o-normal--14-140-75-75-m-90-iso8859-1}
-yscrollcommand {.alarm.frame12.frame.scrollbar2 set}

# build widget .alarm.frame12.frame4
frame .alarm.frame12.frame4 -height {30} -width {30}

# build widget .alarm.frame12.frame4.label5
label .alarm.frame12.frame4.label5 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal--14-140-75-75-p-82-iso8859-1} -foreground {white}
-relief {raised} -width {2}

# build widget .alarm.frame12.frame4.label6
label .alarm.frame12.frame4.label6 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {Clear Time}

# build widget .alarm.frame12.frame4.label8
label .alarm.frame12.frame4.label8 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {Current Value}

# build widget .alarm.frame12.frame4.label9
label .alarm.frame12.frame4.label9 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {Low} -width {6}

# build widget .alarm.frame12.frame4.label10
label .alarm.frame12.frame4.label10 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {High} -width {6}

```

```

# build widget .alarm.frame12.frame4.label11
label .alarm.frame12.frame4.label11 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {Description}

# build widget .alarm.frame12.message7
message .alarm.frame12.message7 -aspect {1500} -background {mediumaquamarine} -borderwidth {0} -font
{-adobe-helvetica-bold-o-normal-14-140-75-75-p-82-iso8859-1} -justify {center} -padx {6} -pady {3} -text {C
L
E
A
R
E
D}

# pack master .alarm.frame1
pack configure .alarm.frame1.frame4 -fill x
pack configure .alarm.frame1.message7 -fill y -side left
pack configure .alarm.frame1.frame -expand 1 -fill both

# pack master .alarm.frame1.frame
pack configure .alarm.frame1.frame.scrollbar2 -fill y -side right
pack configure .alarm.frame1.frame.listbox1 -expand 1 -fill both

# pack master .alarm.frame1.frame4
pack configure .alarm.frame1.frame4.label5 -ipadx 2 -side left
pack configure .alarm.frame1.frame4.label6 -side left
pack configure .alarm.frame1.frame4.label8 -side left
pack configure .alarm.frame1.frame4.label9 -side left
pack configure .alarm.frame1.frame4.label10 -side left
pack configure .alarm.frame1.frame4.label11 -expand 1 -fill x -side left

# pack master .alarm.frame3
pack configure .alarm.frame3.checkbutton0 -fill y -side right
pack configure .alarm.frame3.button15 -side right
pack configure .alarm.frame3.button14 -side right

# pack master .alarm.frame12
pack configure .alarm.frame12.frame4 -fill x
pack configure .alarm.frame12.message7 -fill y -side left
pack configure .alarm.frame12.frame -expand 1 -fill both

# pack master .alarm.frame12.frame
pack configure .alarm.frame12.frame.scrollbar2 -fill y -side right
pack configure .alarm.frame12.frame.listbox1 -expand 1 -fill both

# pack master .alarm.frame12.frame4
pack configure .alarm.frame12.frame4.label5 -ipadx 2 -side left
pack configure .alarm.frame12.frame4.label6 -side left
pack configure .alarm.frame12.frame4.label8 -side left
pack configure .alarm.frame12.frame4.label9 -side left
pack configure .alarm.frame12.frame4.label10 -side left
pack configure .alarm.frame12.frame4.label11 -expand 1 -fill x -side left

# pack master .alarm
pack configure .alarm.frame1 -expand 1 -fill both
pack configure .alarm.frame12 -expand 1 -fill both
pack configure .alarm.frame3 -fill x

if {[info procs XFEdit] != ""} {
  catch "XFMiscBindWidgetTree .alarm"
  after 2 "catch {XFEditSetShowWindows}"
}
}

proc DestroyWindow.alarm {} {# xf ignore me 7
  if {[info procs XFEdit] != ""} {
    if {[info commands .alarm] != ""} {
      global xfShowWindow.alarm
      set xfShowWindow.alarm 0
      XFEditSetPath .
      after 2 "XFSaveAsProc .alarm; XFEditSetShowWindows"
    }
  }
}
}
catch "destroy .alarm"
update

```



```

}
}

# procedure to show window ShowWindow.appmsg
proc ShowWindow.appmsg { args } {
# xf ignore me 7

# build widget .appmsg
if {"[info procs XFedit]" != ""} {
    catch "XFDestroy .appmsg"
} {
    catch "destroy .appmsg"
}
toplevel .appmsg -relief {raised}

# Window manager configurations
wm positionfrom .appmsg program
wm sizefrom .appmsg program
wm maxsize .appmsg 1000 864
wm minsize .appmsg 330 235
wm protocol .appmsg WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .appmsg {Automated Procedure Messages}

# build widget .appmsg.frame
frame .appmsg.frame -relief {raised}

# build widget .appmsg.frame.scrollbar1
scrollbar .appmsg.frame.scrollbar1 -borderwidth {1} -command { .appmsg.frame.text2 yview } -highlightthickness {0} -relief {raised}

# build widget .appmsg.frame.text2
text .appmsg.frame.text2 -background {#c8c8c8} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1} -height {6} -width {40} -wrap
{none} -yscrollcommand { .appmsg.frame.scrollbar1 set }

# build widget .appmsg.frame36
frame .appmsg.frame36 -height {30} -width {30}

# build widget .appmsg.frame36.button7
button .appmsg.frame36.button7 -borderwidth {1} -command { .appmsg.frame.text2 delete 1.0 end } -font
{-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -text {Clear Messages}

# build widget .appmsg.frame36.button37
button .appmsg.frame36.button37 -borderwidth {1} -command {DestroyWindow.appmsg} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx
{11} -pady {4} -text {Done}

# build widget .appmsg.frame36.menubutton38
menubutton .appmsg.frame36.menubutton38 -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -highlightthickness {2} -menu
{ .appmsg.frame36.menubutton38.m } -padx {5} -pady {4} -relief {raised} -text {Font}

# build widget .appmsg.frame36.menubutton38.m
menu .appmsg.frame36.menubutton38.m
.appmsg.frame36.menubutton38.m add radiobutton -command { .appmsg.frame.text2 configure -font -adobe-helvetica-bold-o-normal--12-120-75-75-p-69-iso8859-1 } -font
{ .appmsg.frame.text2 configure -font -adobe-helvetica-bold-o-normal--12-120-75-75-p-69-iso8859-1 } -label {12 point} -value {12} -variable {appmsgfont}
.appmsg.frame36.menubutton38.m add radiobutton -command { .appmsg.frame.text2 configure -font -adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1 }
-font { .appmsg.frame.text2 configure -font -adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1 } -label {18 point} -value {18} -variable {appmsgfont}
.appmsg.frame36.menubutton38.m add radiobutton -command { .appmsg.frame.text2 configure -font -adobe-helvetica-medium-r-normal--34-240-100-100-p-176-iso8859-1 }
-font { .appmsg.frame.text2 configure -font -adobe-helvetica-medium-r-normal--34-240-100-100-p-176-iso8859-1 } -label {34 point} -value {34} -variable {appmsgfont}

# pack master .appmsg.frame
pack configure .appmsg.frame.scrollbar1 -fill y -side right
pack configure .appmsg.frame.text2 -expand 1 -fill both

# pack master .appmsg.frame36
pack configure .appmsg.frame36.menubutton38 -fill y -side right
pack configure .appmsg.frame36.button7 -side right
pack configure .appmsg.frame36.button37 -side right

# pack master .appmsg
pack configure .appmsg.frame -expand 1 -fill both
pack configure .appmsg.frame36 -fill x

.appmsg.frame.text2 insert end {}

```

```

if {[info procs XFEdit] != ""} {
  catch "XFMiscBindWidgetTree .appmsg"
  after 2 "catch {XFEditSetShowWindows}"
}
}

```

```

proc DestroyWindow.appmsg {} {# xf ignore me 7
if {[info procs XFEdit] != ""} {
  if {[info commands .appmsg] != ""} {
    global xfShowWindow.appmsg
    set xfShowWindow.appmsg 0
    XFEditSetPath .
    after 2 "XFSaveAsProc .appmsg; XFEditSetShowWindows"
  }
} {
  catch "destroy .appmsg"
  update
}
}

```

procedure to show window ShowWindow.aps

```

proc ShowWindow.aps { args } {
# xf ignore me 7

```

```

# build widget .aps
if {[info procs XFEdit] != ""} {
  catch "XFDestroy .aps"
} {
  catch "destroy .aps"
}
}
toplevel .aps -relief {raised}

```

```

# Window manager configurations
wm positionfrom .aps user
wm sizefrom .aps program
wm geometry .aps +37+143
wm maxsize .aps 1152 864
wm minsize .aps 562 539
wm protocol .aps WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .aps {Automated Procedures}

```

```

# build widget .aps.frame
frame .aps.frame

```

```

# build widget .aps.frame.scrollbar3
scrollbar .aps.frame.scrollbar3 -borderwidth {1} -command {aps.frame.listbox1 xview} -orient {horizontal} -relief {raised}

```

```

# build widget .aps.frame.scrollbar2
scrollbar .aps.frame.scrollbar2 -borderwidth {1} -command {aps.frame.listbox1 yview} -relief {raised}

```

```

# build widget .aps.frame.listbox1
listbox .aps.frame.listbox1 -background {#c8dcdc} -borderwidth {1} -highlightthickness {0} -xscrollcommand {aps.frame.scrollbar3 set} -yscrollcommand
{aps.frame.scrollbar2 set}

```

```

# build widget .aps.frame1
frame .aps.frame1

```

```

# build widget .aps.frame1.scrollbar3
scrollbar .aps.frame1.scrollbar3 -borderwidth {1} -command {aps.frame1.listbox1 xview} -orient {horizontal} -relief {raised}

```

```

# build widget .aps.frame1.scrollbar2
scrollbar .aps.frame1.scrollbar2 -borderwidth {1} -command {aps.frame1.listbox1 yview} -relief {raised}

```

```

# build widget .aps.frame1.listbox1
listbox .aps.frame1.listbox1 -background {#c8dcdc} -borderwidth {1} -highlightthickness {0} -xscrollcommand {aps.frame1.scrollbar3 set} -yscrollcommand
{aps.frame1.scrollbar2 set}

```

```

# build widget .aps.frame2
frame .aps.frame2 -borderwidth {2} -height {30} -relief {ridge} -width {30}

```

```

# build widget .aps.frame2.label5
label .aps.frame2.label5 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white} -relief
{raised} -text { Start Time }

# build widget .aps.frame2.label6
label .aps.frame2.label6 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white} -relief
{raised} -text { Stop Time }

# build widget .aps.frame2.label7
label .aps.frame2.label7 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white} -relief
{raised} -text { Status }

# build widget .aps.frame2.label8
label .aps.frame2.label8 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white} -relief
{raised} -text {Command}

# build widget .aps.frame3
frame .aps.frame3 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .aps.frame3.label0
label .aps.frame3.label0 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white} -relief
{raised} -text {Procedure Name}

# build widget .aps.frame3.label1
label .aps.frame3.label1 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white} -relief
{raised} -text {Description}

# build widget .aps.frame4
frame .aps.frame4 -height {30} -width {30}

# build widget .aps.frame4.button2
button .aps.frame4.button2 -background {darkseagreen} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4}
-text {Skip}

# build widget .aps.frame4.button3
button .aps.frame4.button3 -background {lightsteelblue} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4}
-text {Pause}

# build widget .aps.frame4.button4
button .aps.frame4.button4 -background {lightsteelblue} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4}
-text {Resume}

# build widget .aps.frame4.button5
button .aps.frame4.button5 -background {lightsteelblue} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4}
-text {Terminate}

# build widget .aps.frame4.button6
button .aps.frame4.button6 -background {lightsteelblue} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4}
-text {Start}

# build widget .aps.frame4.button7
button .aps.frame4.button7 -borderwidth {1} -command {DestroyWindow.aps} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady
{4} -text {Done}

# build widget .aps.frame4.button8
button .aps.frame4.button8 -background {darkseagreen} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4}
-text {Execute}

# pack master .aps.frame
pack configure .aps.frame.scrollbar2 -fill y -side right
pack configure .aps.frame.listbox1 -expand 1 -fill both
pack configure .aps.frame.scrollbar3 -fill x -side bottom

# pack master .aps.frame1
pack configure .aps.frame1.scrollbar2 -fill y -side right
pack configure .aps.frame1.listbox1 -expand 1 -fill both
pack configure .aps.frame1.scrollbar3 -fill x -side bottom

# pack master .aps.frame2
pack configure .aps.frame2.label5 -side left
pack configure .aps.frame2.label6 -side left
pack configure .aps.frame2.label7 -side left
pack configure .aps.frame2.label8 -expand 1 -fill x -side left

# pack master .aps.frame3

```

```

pack configure .aps.frame3.label0 -side left
pack configure .aps.frame3.label1 -expand 1 -fill x -side left

# pack master .aps.frame4
pack configure .aps.frame4.button2 -expand 1 -fill x -side left
pack configure .aps.frame4.button8 -expand 1 -fill x -side left
pack configure .aps.frame4.button3 -expand 1 -fill x -side left
pack configure .aps.frame4.button4 -expand 1 -fill x -side left
pack configure .aps.frame4.button5 -expand 1 -fill x -side left
pack configure .aps.frame4.button6 -expand 1 -fill x -side left
pack configure .aps.frame4.button7 -expand 1 -fill x -side left

# pack master .aps
pack configure .aps.frame2 -fill x
pack configure .aps.frame -expand 1 -fill both
pack configure .aps.frame3 -fill x
pack configure .aps.frame1 -expand 1 -fill both
pack configure .aps.frame4 -fill x

if {"[info procs XFEdit]" != ""} {
  catch "XFMiscBindWidgetTree .aps"
  after 2 "catch {XFEditSetShowWindows}"
}
}

proc DestroyWindow.aps {} {# xf ignore me 7
if {"[info procs XFEdit]" != ""} {
  if {"[info commands .aps]" != ""} {
    global xfShowWindow.aps
    set xfShowWindow.aps 0
    XFEditSetPath .
    after 2 "XFSaveAsProc .aps; XFEditSetShowWindows"
  }
} {
  catch "destroy .aps"
  update
}
}

# procedure to show window ShowWindow.backtrack
proc ShowWindow.backtrack { args } {
# xf ignore me 7

# build widget .backtrack
if {"[info procs XFEdit]" != ""} {
  catch "XFDestroy .backtrack"
} {
  catch "destroy .backtrack"
}
}
toplevel .backtrack -background {grey65} -relief {raised}

# Window manager configurations
wm positionfrom .backtrack program
wm sizefrom .backtrack program
wm maxsize .backtrack 1152 864
wm minsize .backtrack 543 310
wm protocol .backtrack WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .backtrack {BackTrack / Replay}

# build widget .backtrack.frame
frame .backtrack.frame

# build widget .backtrack.frame.scrollbar2
scrollbar .backtrack.frame.scrollbar2 -borderwidth {1} -command {backtrack.frame.listbox1 yview} -highlightthickness {0} -relief {raised}

# build widget .backtrack.frame.listbox1
listbox .backtrack.frame.listbox1 -background {#c8dcdc} -borderwidth {1} -font {-adobe-courier-bold-o-normal-14-140-75-75-m-90-iso8859-1} -yscrollcommand
{backtrack.frame.scrollbar2 set}

# build widget .backtrack.frame.frame0
frame .backtrack.frame.frame0 -borderwidth {2} -height {30} -relief {ridge} -width {30}

```

```

# build widget .backtrack.frame.frame0.label1
label .backtrack.frame.frame0.label1 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -foreground {white}
-relief {raised} -text {No.} -width {6}

# build widget .backtrack.frame.frame0.label2
label .backtrack.frame.frame0.label2 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -foreground {white}
-relief {raised} -text {Date} -width {10}

# build widget .backtrack.frame.frame0.label3
label .backtrack.frame.frame0.label3 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -foreground {white}
-relief {raised} -text {Real Time} -width {13}

# build widget .backtrack.frame.frame0.label4
label .backtrack.frame.frame0.label4 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -foreground {white}
-relief {raised} -text {Load Name} -width {16}

# build widget .backtrack.frame.frame0.label0
label .backtrack.frame.frame0.label0 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -foreground {white}
-relief {raised} -text {Virtual Time} -width {14}

# build widget .backtrack.frame21
frame .backtrack.frame21 -background {grey65} -height {30} -width {30}

# build widget .backtrack.frame21.label20
label .backtrack.frame21.label20 -background {grey65} -borderwidth {0} -font {-adobe-times-bold-i-normal--14-140-75-75-p-77-iso8859-1} -text {Backtrack interval (secs):}

# build widget .backtrack.frame21.frame0
frame .backtrack.frame21.frame0 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .backtrack.frame21.frame0.label24
label .backtrack.frame21.frame0.label24 -background {grey65} -borderwidth {0} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -text {Current:}

# build widget .backtrack.frame21.frame0.label22
label .backtrack.frame21.frame0.label22 -background {grey65} -borderwidth {0} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -foreground {red}
-text {60} -textvariable {bt_interval} -width {5}

# build widget .backtrack.frame21.frame3
frame .backtrack.frame21.frame3 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .backtrack.frame21.frame3.label24
label .backtrack.frame21.frame3.label24 -background {grey65} -borderwidth {0} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -text {Requested:}

# build widget .backtrack.frame21.frame3.entry25
entry .backtrack.frame21.frame3.entry25 -background {lightblue} -borderwidth {1} -font {-adobe-helvetica-medium-o-normal--12-120-75-75-p-67-iso8859-1}
-highlightthickness {0} -textvariable {bt_req_interval} -width {6}

# build widget .backtrack.frame21.frame3.button4
button .backtrack.frame21.frame3.button4 -borderwidth {1} -command {css_cmd ctrl "bt_int_set $bt_req_interval"} -font
{-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -highlightthickness {1} -padx {6} -pady {2} -text {Set Interval}

# build widget .backtrack.frame5
frame .backtrack.frame5 -background {grey65} -height {30} -width {30}

# build widget .backtrack.frame5.button0
button .backtrack.frame5.button0 -borderwidth {1} -command {DestroyWindow.backtrack} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1}
-highlightbackground {grey65} -padx {11} -pady {4} -text {Done}

# build widget .backtrack.frame5.button26
button .backtrack.frame5.button26 -borderwidth {1} -command {set index [.backtrack.frame.listbox1 curselection]
if {$index != ""}{
    set bcsselected [lindex [.backtrack.frame.listbox1 get $index] 0]
    if {$ics_confirm == "on"} {
        set ans [tk_messageBox -icon question -type yesno -message "Do you really want to RESET into BACKTRACK condition $bcsselected?"]
        } {set ans yes}

    if {$ans == "yes"} {
        if {[string index $curlC 0] == "I"} {
            css_cmd ctrl "snap IC00"
            set bt_origC $curlC
            set bt_totsecs [css_cmd ctrl "getval seconds"]
            set bt_totframes [css_cmd ctrl "getval totframes"]
        }
        css_cmd ctrl "bt_reset BC$bcsselected"
    }
}
} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -highlightbackground {grey65} -padx {11} -pady {4} -text {Reset to}

```

```

# build widget .backtrack.frame5.button15
button .backtrack.frame5.button15 -borderwidth {1} -command {css_cmd ctrl "reset IC00 $bl_orgIC"}
after 100
css_cmd ctrl "setval seconds $bl_totsecs"
css_cmd ctrl "setval totframes $bl_totframes"
DestroyWindow.backtrack} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -highlightbackground {grey65} -padx {11} -pady {4} -state {active}
-text {Cancel}

# build widget .backtrack.frame10
frame .backtrack.frame10 -background {grey65} -height {30} -width {30}

# build widget .backtrack.frame10.label11
label .backtrack.frame10.label11 -background {grey65} -borderwidth {0} -font {-adobe-times-bold-i-normal--14-140-75-75-p-77-iso8859-1} -text {Replay;}

# build widget .backtrack.frame10.radiobutton12
radiobutton .backtrack.frame10.radiobutton12 -background {grey65} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -highlightthickness {0} -text
{Stop} -value {stop} -variable {replay_sta}

# build widget .backtrack.frame10.radiobutton13
radiobutton .backtrack.frame10.radiobutton13 -background {grey65} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -highlightthickness {0} -text
{Run} -value {run} -variable {replay_sta}

# build widget .backtrack.frame10.radiobutton14
radiobutton .backtrack.frame10.radiobutton14 -background {grey65} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -highlightthickness {0} -text
{Pause} -value {pause} -variable {replay_sta}

# pack master .backtrack.frame
pack configure .backtrack.frame.scrollbar2 -fill y -side right
pack configure .backtrack.frame.frame0 -fill x
pack configure .backtrack.frame.listbox1 -expand 1 -fill both

# pack master .backtrack.frame.frame0
pack configure .backtrack.frame.frame0.label1 -side left
pack configure .backtrack.frame.frame0.label2 -side left
pack configure .backtrack.frame.frame0.label3 -side left
pack configure .backtrack.frame.frame0.label0 -side left
pack configure .backtrack.frame.frame0.label4 -expand 1 -fill x -side left

# pack master .backtrack.frame21
pack configure .backtrack.frame21.label20 -side left
pack configure .backtrack.frame21.frame0 -fill y -side left
pack configure .backtrack.frame21.frame3 -side left

# pack master .backtrack.frame21.frame0
pack configure .backtrack.frame21.frame0.label24 -expand 1 -fill both -side left
pack configure .backtrack.frame21.frame0.label22 -expand 1 -fill both -side left

# pack master .backtrack.frame21.frame3
pack configure .backtrack.frame21.frame3.label24 -expand 1 -fill both -side left
pack configure .backtrack.frame21.frame3.entry25 -expand 1 -fill y -side left
pack configure .backtrack.frame21.frame3.button4 -expand 1 -fill x -side left

# pack master .backtrack.frame5
pack configure .backtrack.frame5.button0 -expand 1 -fill x -side left
pack configure .backtrack.frame5.button26 -expand 1 -fill x -side left
pack configure .backtrack.frame5.button15 -expand 1 -fill x -side left

# pack master .backtrack.frame10
pack configure .backtrack.frame10.label11 -side left
pack configure .backtrack.frame10.radiobutton12 -padx 6 -side left
pack configure .backtrack.frame10.radiobutton13 -padx 6 -side left
pack configure .backtrack.frame10.radiobutton14 -padx 6 -side left

# pack master .backtrack
pack configure .backtrack.frame -expand 1 -fill both
pack configure .backtrack.frame21 -fill x
pack configure .backtrack.frame5 -anchor se -pady 2 -side bottom
pack configure .backtrack.frame10 -fill x

.backtrack.frame.listbox1 insert end { 00 10/02/97 20:30:46 01:09:18 rtsched.exe}
.backtrack.frame.listbox1 insert end { 01 10/02/97 20:30:52 01:09:24 rtsched.exe}
.backtrack.frame.listbox1 insert end { 02 10/02/97 20:30:58 01:09:30 rtsched.exe}
.backtrack.frame.listbox1 insert end { 03 10/02/97 20:31:04 01:09:36 rtsched.exe}
.backtrack.frame.listbox1 insert end { 04 10/02/97 20:31:11 01:09:42 rtsched.exe}

```

```

.backtrack.frame.listbox1 insert end { 05 10/02/97 20:31:17 01:09:48 rtsched.exe}
.backtrack.frame.listbox1 insert end { 06 10/02/97 20:31:23 01:09:54 rtsched.exe}
.backtrack.frame.listbox1 insert end { 07 10/02/97 20:31:29 01:10:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 08 10/02/97 20:31:35 01:10:06 rtsched.exe}
.backtrack.frame.listbox1 insert end { 09 10/02/97 20:31:40 01:10:12 rtsched.exe}
.backtrack.frame.listbox1 insert end { 10 10/02/97 20:31:46 01:10:18 rtsched.exe}
.backtrack.frame.listbox1 insert end { 11 10/02/97 20:31:52 01:10:24 rtsched.exe}
.backtrack.frame.listbox1 insert end { 12 10/02/97 20:31:58 01:10:30 rtsched.exe}
.backtrack.frame.listbox1 insert end { 13 10/02/97 20:32:04 01:10:36 rtsched.exe}
.backtrack.frame.listbox1 insert end { 14 10/02/97 20:32:10 01:10:42 rtsched.exe}
.backtrack.frame.listbox1 insert end { 15 10/02/97 20:32:16 01:10:48 rtsched.exe}
.backtrack.frame.listbox1 insert end { 16 10/02/97 20:32:22 01:10:54 rtsched.exe}
.backtrack.frame.listbox1 insert end { 17 10/02/97 20:32:28 01:11:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 18 10/02/97 20:32:34 01:11:06 rtsched.exe}
.backtrack.frame.listbox1 insert end { 19 10/02/97 20:32:40 01:11:12 rtsched.exe}
.backtrack.frame.listbox1 insert end { 20 10/02/97 20:32:46 01:11:19 rtsched.exe}
.backtrack.frame.listbox1 insert end { 21 10/02/97 20:32:53 01:11:25 rtsched.exe}
.backtrack.frame.listbox1 insert end { 22 10/02/97 20:32:59 01:11:31 rtsched.exe}
.backtrack.frame.listbox1 insert end { 23 10/02/97 20:33:05 01:11:37 rtsched.exe}
.backtrack.frame.listbox1 insert end { 24 10/02/97 20:33:11 01:11:43 rtsched.exe}
.backtrack.frame.listbox1 insert end { 25 10/02/97 20:33:17 01:11:49 rtsched.exe}
.backtrack.frame.listbox1 insert end { 26 10/02/97 20:33:23 01:11:55 rtsched.exe}
.backtrack.frame.listbox1 insert end { 27 10/02/97 20:33:29 01:12:01 rtsched.exe}
.backtrack.frame.listbox1 insert end { 28 10/02/97 20:33:35 01:12:07 rtsched.exe}
.backtrack.frame.listbox1 insert end { 29 10/02/97 20:33:41 01:12:13 rtsched.exe}
.backtrack.frame.listbox1 insert end { 30 10/02/97 20:33:47 01:12:19 rtsched.exe}
.backtrack.frame.listbox1 insert end { 31 10/02/97 20:33:53 01:12:25 rtsched.exe}
.backtrack.frame.listbox1 insert end { 32 10/02/97 20:33:59 01:12:31 rtsched.exe}
.backtrack.frame.listbox1 insert end { 33 10/02/97 20:34:05 01:12:37 rtsched.exe}
.backtrack.frame.listbox1 insert end { 34 10/02/97 20:34:11 01:12:43 rtsched.exe}
.backtrack.frame.listbox1 insert end { 35 10/02/97 20:34:17 01:12:49 rtsched.exe}
.backtrack.frame.listbox1 insert end { 36 10/02/97 20:34:23 01:12:55 rtsched.exe}
.backtrack.frame.listbox1 insert end { 37 10/02/97 20:34:29 01:13:01 rtsched.exe}
.backtrack.frame.listbox1 insert end { 38 10/02/97 20:34:35 01:13:07 rtsched.exe}
.backtrack.frame.listbox1 insert end { 39 10/02/97 20:34:41 01:13:13 rtsched.exe}
.backtrack.frame.listbox1 insert end { 40 10/03/97 18:38:44 00:00:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 41 10/03/97 18:39:55 00:00:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 42 10/03/97 18:40:48 00:00:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 43 10/03/97 18:41:22 00:00:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 44 10/03/97 18:45:39 01:12:31 rtsched.exe}
.backtrack.frame.listbox1 insert end { 45 10/03/97 18:45:56 01:08:54 rtsched.exe}
.backtrack.frame.listbox1 insert end { 46 10/03/97 18:46:55 01:09:54 rtsched.exe}
.backtrack.frame.listbox1 insert end { 47 10/03/97 18:47:55 01:10:54 rtsched.exe}
.backtrack.frame.listbox1 insert end { 48 10/03/97 18:48:56 01:11:54 rtsched.exe}
.backtrack.frame.listbox1 insert end { 49 10/03/97 18:49:55 01:12:54 rtsched.exe}
.backtrack.frame.listbox1 insert end { 50 10/03/97 18:50:55 01:13:54 rtsched.exe}
.backtrack.frame.listbox1 insert end { 51 10/03/97 18:51:19 00:00:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 52 10/03/97 18:52:59 01:10:54 rtsched.exe}
.backtrack.frame.listbox1 insert end { 53 10/03/97 18:54:48 01:12:01 rtsched.exe}
.backtrack.frame.listbox1 insert end { 54 10/03/97 19:00:24 00:00:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 55 10/03/97 19:01:24 00:01:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 56 10/03/97 19:02:23 00:02:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 57 10/03/97 19:03:23 00:03:00 rtsched.exe}
.backtrack.frame.listbox1 insert end { 58 10/03/97 19:04:04 01:12:19 rtsched.exe}
.backtrack.frame.listbox1 insert end { 59 10/02/97 20:30:40 01:09:12 rtsched.exe}

```

```

if {"[info procs XFEdit]" != ""} {
  catch "XFMiscBindWidgetTree .backtrack"
  after 2 "catch {XFEditSetShowWindows}"
}

```

```

proc DestroyWindow.backtrack {} {# xf ignore me 7
  if {"[info procs XFEdit]" != ""} {
    if {"[info commands .backtrack]" != ""} {
      global xfShowWindow.backtrack
      set xfShowWindow.backtrack 0
      XFEditSetPath .
      after 2 "XFSaveAsProc .backtrack; XFEditSetShowWindows"
    }
  }
  catch "destroy .backtrack"
  update
}

```

```

}
}

# procedure to show window ShowWindow.diagidx
proc ShowWindow.diagidx { args } {
# xf ignore me 7

# build widget .diagidx
if {[info procs XFEdit] != ""} {
  catch "XFDestroy .diagidx"
}{
  catch "destroy .diagidx"
}
toplevel .diagidx

# Window manager configurations
wm positionfrom .diagidx program
wm sizelfrom .diagidx program
wm maxsize .diagidx 1028 753
wm minsize .diagidx 104 1
wm protocol .diagidx WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .diagidx {Diagram Index}

# bindings
bind .diagidx <Reparent> {set cwd [pwd]}
cd Data
set flist [glob *.tcl]
cd $cwd
.diagidx.frame.listbox1 delete 0 end
eval ".diagidx.frame.listbox1 insert end $flist"
}

# build widget .diagidx.frame
frame .diagidx.frame

# build widget .diagidx.frame.scrollbar3
scrollbar .diagidx.frame.scrollbar3 -borderwidth {1} -command {.diagidx.frame.listbox1 xview} -orient {horizontal} -relief {raised}

# build widget .diagidx.frame.scrollbar2
scrollbar .diagidx.frame.scrollbar2 -activerelief {sunken} -borderwidth {1} -command {.diagidx.frame.listbox1 yview} -relief {raised}

# build widget .diagidx.frame.listbox1
listbox .diagidx.frame.listbox1 -background {#c8dcdc} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal-14-140-75-75-p-82-iso8859-1} -highlightthickness {0}
-xscrollcommand {.diagidx.frame.scrollbar3 set} -yscrollcommand {.diagidx.frame.scrollbar2 set}
# bindings
bind .diagidx.frame.listbox1 <Button-1> {set index [%W nearest %y]}
set dragbuffer [%W get $index]
show.dragwin %X %Y
set icon_on 1
move_icon}
bind .diagidx.frame.listbox1 <ButtonRelease-1> {set icon_on 0
drop.dragwin}

# build widget .diagidx.frame1
frame .diagidx.frame1 -height {30} -width {30}

# build widget .diagidx.frame1.button2
button .diagidx.frame1.button2 -borderwidth {1} -command {vcs_show} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4}
-text {Show VCS}

# build widget .diagidx.frame1.button3
button .diagidx.frame1.button3 -borderwidth {1} -command {DestroyWindow.diagidx} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11}
-pady {4} -text {Done}

# pack master .diagidx.frame
pack configure .diagidx.frame.scrollbar2 -fill y -side right
pack configure .diagidx.frame.listbox1 -expand 1 -fill both
pack configure .diagidx.frame.scrollbar3 -fill x -side bottom

# pack master .diagidx.frame1
pack configure .diagidx.frame1.button2 -side right
pack configure .diagidx.frame1.button3 -side right

```



```

# pack master .diagidx
pack configure .diagidx.frame -expand 1 -fill both
pack configure .diagidx.frame1 -fill x

.diagidx.frame.listbox1 insert end {ed1.tcl}
.diagidx.frame.listbox1 insert end {ed2.tcl}

if {"[info procs XFEdit]" != ""} {
    catch "XFMiscBindWidgetTree .diagidx" -
    after 2 "catch {XFEditSetShowWindows}"
}

proc DestroyWindow.diagidx {} {# xf ignore me 7
if {"[info procs XFEdit]" != ""} {
    if {"[info commands .diagidx]" != ""} {
        global xfShowWindow.diagidx
        set xfShowWindow.diagidx 0
        XFEditSetPath .
        after 2 "XFSaveAsProc .diagidx; XFEditSetShowWindows"
    }
} {
    catch "destroy .diagidx"
    update
}
}

# procedure to show window ShowWindow.dispstat
proc ShowWindow.dispstat { args } {
# xf ignore me 7

# build widget .dispstat
if {"[info procs XFEdit]" != ""} {
    catch "XFDestroy .dispstat"
} {
    catch "destroy .dispstat"
}
}
toplevel .dispstat -relief {raised}

# Window manager configurations
wm positionfrom .dispstat program
wm sizefrom .dispstat program
wm geometry .dispstat 363x288
wm maxsize .dispstat 363 600
wm minsize .dispstat 217 211
wm protocol .dispstat WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path." to show/hide windows.}}
wm title .dispstat {Display Status}

# build widget .dispstat.label1
label .dispstat.label1 -background {steelblue} -font {adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white} -relief {raised} -text
{Connections from asus} -textvariable {stathost}

# build widget .dispstat.frame
frame .dispstat.frame

# build widget .dispstat.frame.scrollbar2
scrollbar .dispstat.frame.scrollbar2 -command {.dispstat.frame.listbox1 yview} -relief {raised}

# build widget .dispstat.frame.listbox1
listbox .dispstat.frame.listbox1 -borderwidth {1} -font {adobe-courier-bold-r-normal-12-120-75-75-m-70-iso8859-1} -height {6} -yscrollcommand
{.dispstat.frame.scrollbar2 set}

# build widget .dispstat.frame.frame1
frame .dispstat.frame.frame1 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .dispstat.frame.frame1.label2
label .dispstat.frame.frame1.label2 -background {slategrey} -borderwidth {1} -font {adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {Name} -width {16}

# build widget .dispstat.frame.frame1.label3

```

```

label .dispstat.frame.frame1.label3 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {white}
-relief {raised} -text {Connection Established} -width {22}

# build widget .dispstat.frame2
frame .dispstat.frame2 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .dispstat.frame2.button3
button .dispstat.frame2.button3 -command {DestroyWindow.dispstat} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -state
{active} -text {Done}

# pack master .dispstat.frame
pack configure .dispstat.frame.scrollbar2 -fill y -side right
pack configure .dispstat.frame.frame1 -fill x
pack configure .dispstat.frame.listbox1 -expand 1 -fill both

# pack master .dispstat.frame.frame1
pack configure .dispstat.frame.frame1.label2 -side left
pack configure .dispstat.frame.frame1.label3 -side left

# pack master .dispstat.frame2
pack configure .dispstat.frame2.button3 -expand 1 -fill x -side left

# pack master .dispstat
pack configure .dispstat.label1 -fill x
pack configure .dispstat.frame -expand 1 -fill both
pack configure .dispstat.frame2 -fill x

.dispstat.frame.listbox1 insert end {cserver.tcl 09/21/97 20:06:06}
.dispstat.frame.listbox1 insert end {xfmain.tcl 09/21/97 20:06:15}

if {[info procs XFEdit] != ""} {
  catch *XFMiscBindWidgetTree .dispstat
  after 2 *catch {XFEditSetShowWindows}
}
}

proc DestroyWindow.dispstat {} {# xf ignore me 7
if {[info procs XFEdit] != ""} {
  if {[info commands .dispstat] != ""} {
    global xfShowWindow.dispstat
    set xfShowWindow.dispstat 0
    XFEditSetPath .
    after 2 *XFSaveAsProc .dispstat; XFEditSetShowWindows"
  }
} {
  catch "destroy .dispstat"
  update
}
}

# procedure to show window ShowWindow.ets
proc ShowWindow.ets { args } {
# xf ignore me 7

# build widget .ets
if {[info procs XFEdit] != ""} {
  catch *XFDestroy .ets"
} {
  catch "destroy .ets"
}
}
toplevel .ets -relief {raised}

# Window manager configurations
wm positionfrom .ets program
wm sizefrom .ets program
wm maxsize .ets 1000 864
wm minsize .ets 104 10
wm protocol .ets WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .ets {Event Triggers}

```

```

# build widget .ets.frame10
frame .ets.frame10 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .ets.frame10.label11
label .ets.frame10.label11 -background {steelblue} -font {-adobe-helvetica-bold-o-normal--14-140-75-75-p-82-iso8859-1} -foreground {white} -relief {raised} -text {Trigger}
-width {7}

# build widget .ets.frame10.label12
label .ets.frame10.label12 -background {steelblue} -font {-adobe-helvetica-bold-o-normal--14-140-75-75-p-82-iso8859-1} -foreground {white} -relief {raised} -text {File}
-width {13}

# build widget .ets.frame10.label13
label .ets.frame10.label13 -background {steelblue} -font {-adobe-helvetica-bold-o-normal--14-140-75-75-p-82-iso8859-1} -foreground {white} -relief {raised} -text
{Description} -width {20}

# build widget .ets.frame10.label17
label .ets.frame10.label17 -background {steelblue} -font {-adobe-helvetica-bold-o-normal--14-140-75-75-p-82-iso8859-1} -foreground {white} -relief {raised} -text {Status}
-width {6}

# build widget .ets.frame10.label23
label .ets.frame10.label23 -background {steelblue} -foreground {white} -relief {raised} -text {Control} -width {21}

# build widget .ets.frame14
frame .ets.frame14 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .ets.frame14.button18
button .ets.frame14.button18 -background {slategrey} -command {DestroyWindow.ets} -font {-adobe-helvetica-medium-o-normal--18-180-75-75-p-98-iso8859-1}
-foreground {green} -padx {11} -pady {4} -state {active} -text {Done}

# build widget .ets.frame0
frame .ets.frame0 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .ets.frame0.frame10
frame .ets.frame0.frame10 -height {30} -width {30}

# build widget .ets.frame0.frame10.label11
label .ets.frame0.frame10.label11 -background {darkseagreen} -relief {raised} -text {E1} -width {9}

# build widget .ets.frame0.frame10.label13
label .ets.frame0.frame10.label13 -background {darkseagreen} -relief {raised} -textvariable {et_desc(1)}

# build widget .ets.frame0.frame10.label17
label .ets.frame0.frame10.label17 -background {mediumaquamarine} -relief {raised} -text {0} -textvariable {et(1)} -width {8}

# build widget .ets.frame0.frame10.entry13
entry .ets.frame0.frame10.entry13 -background {mediumturquoise} -font {-adobe-helvetica-medium-o-normal--12-120-75-75-p-67-iso8859-1} -relief {raised} -textvariable
{et_frame(1)} -width {14}

# build widget .ets.frame0.frame10.radiobutton27
radiobutton .ets.frame0.frame10.radiobutton27 -text {ON} -value {ON} -variable {et_ctrl(1)}

# build widget .ets.frame0.frame10.radiobutton28
radiobutton .ets.frame0.frame10.radiobutton28 -text {OFF} -value {OFF} -variable {et_ctrl(1)}

# build widget .ets.frame0.frame10.radiobutton29
radiobutton .ets.frame0.frame10.radiobutton29 -text {File} -value {FILE} -variable {et_ctrl(1)}

# build widget .ets.frame0.frame14
frame .ets.frame0.frame14 -height {30} -width {30}

# build widget .ets.frame0.frame14.label11
label .ets.frame0.frame14.label11 -background {darkseagreen} -relief {raised} -text {E2} -width {9}

# build widget .ets.frame0.frame14.label13
label .ets.frame0.frame14.label13 -background {darkseagreen} -relief {raised} -textvariable {et_desc(2)}

# build widget .ets.frame0.frame14.label17
label .ets.frame0.frame14.label17 -background {mediumaquamarine} -relief {raised} -text {0} -textvariable {et(2)} -width {8}

# build widget .ets.frame0.frame14.entry13
entry .ets.frame0.frame14.entry13 -background {mediumturquoise} -font {-adobe-helvetica-medium-o-normal--12-120-75-75-p-67-iso8859-1} -relief {raised} -textvariable
{et_frame(2)} -width {14}

# build widget .ets.frame0.frame14.radiobutton27
radiobutton .ets.frame0.frame14.radiobutton27 -text {ON} -value {ON} -variable {et_ctrl(2)}

```

```

# build widget .ets.frame0.frame14.radiobutton28
radiobutton .ets.frame0.frame14.radiobutton28 -text {OFF} -value {OFF} -variable {et_ctrl(2)}

# build widget .ets.frame0.frame14.radiobutton29
radiobutton .ets.frame0.frame14.radiobutton29 -text {File} -value {FILE} -variable {et_ctrl(2)}

# build widget .ets.frame0.frame15
frame .ets.frame0.frame15 -height {30} -width {30}

# build widget .ets.frame0.frame15.label11
label .ets.frame0.frame15.label11 -background {darkseagreen} -relief {raised} -text {E3} -width {9}

# build widget .ets.frame0.frame15.label13
label .ets.frame0.frame15.label13 -background {darkseagreen} -relief {raised} -textvariable {et_desc(3)}

# build widget .ets.frame0.frame15.label17
label .ets.frame0.frame15.label17 -background {mediumaquamarine} -relief {raised} -text {0} -textvariable {et(3)} -width {8}

# build widget .ets.frame0.frame15.entry13
entry .ets.frame0.frame15.entry13 -background {mediumturquoise} -font {adobe-helvetica-medium-o-normal-12-120-75-75-p-67-iso8859-1} -relief {raised} -textvariable {et_frame(3)} -width {14}

# build widget .ets.frame0.frame15.radiobutton27
radiobutton .ets.frame0.frame15.radiobutton27 -text {ON} -value {ON} -variable {et_ctrl(3)}

# build widget .ets.frame0.frame15.radiobutton28
radiobutton .ets.frame0.frame15.radiobutton28 -text {OFF} -value {OFF} -variable {et_ctrl(3)}

# build widget .ets.frame0.frame15.radiobutton29
radiobutton .ets.frame0.frame15.radiobutton29 -text {File} -value {FILE} -variable {et_ctrl(3)}

# build widget .ets.frame0.frame16
frame .ets.frame0.frame16 -height {30} -width {30}

# build widget .ets.frame0.frame16.label11
label .ets.frame0.frame16.label11 -background {darkseagreen} -relief {raised} -text {E4} -width {9}

# build widget .ets.frame0.frame16.label13
label .ets.frame0.frame16.label13 -background {darkseagreen} -relief {raised} -textvariable {et_desc(4)}

# build widget .ets.frame0.frame16.label17
label .ets.frame0.frame16.label17 -background {mediumaquamarine} -relief {raised} -text {0} -textvariable {et(4)} -width {8}

# build widget .ets.frame0.frame16.entry13
entry .ets.frame0.frame16.entry13 -background {mediumturquoise} -font {adobe-helvetica-medium-o-normal-12-120-75-75-p-67-iso8859-1} -relief {raised} -textvariable {et_frame(4)} -width {14}

# build widget .ets.frame0.frame16.radiobutton27
radiobutton .ets.frame0.frame16.radiobutton27 -text {ON} -value {ON} -variable {et_ctrl(4)}

# build widget .ets.frame0.frame16.radiobutton28
radiobutton .ets.frame0.frame16.radiobutton28 -text {OFF} -value {OFF} -variable {et_ctrl(4)}

# build widget .ets.frame0.frame16.radiobutton29
radiobutton .ets.frame0.frame16.radiobutton29 -text {File} -value {FILE} -variable {et_ctrl(4)}

# build widget .ets.frame0.frame17
frame .ets.frame0.frame17 -height {30} -width {30}

# build widget .ets.frame0.frame17.label11
label .ets.frame0.frame17.label11 -background {darkseagreen} -relief {raised} -text {E5} -width {9}

# build widget .ets.frame0.frame17.label13
label .ets.frame0.frame17.label13 -background {darkseagreen} -relief {raised} -textvariable {et_desc(5)}

# build widget .ets.frame0.frame17.label17
label .ets.frame0.frame17.label17 -background {mediumaquamarine} -relief {raised} -text {0} -textvariable {et(5)} -width {8}

# build widget .ets.frame0.frame17.entry13
entry .ets.frame0.frame17.entry13 -background {mediumturquoise} -font {adobe-helvetica-medium-o-normal-12-120-75-75-p-67-iso8859-1} -relief {raised} -textvariable {et_frame(5)} -width {14}

# build widget .ets.frame0.frame17.radiobutton27
radiobutton .ets.frame0.frame17.radiobutton27 -text {ON} -value {ON} -variable {et_ctrl(5)}

```

```
# build widget .ets.frame0.frame17.radiobutton28
radiobutton .ets.frame0.frame17.radiobutton28 -text {OFF} -value {OFF} -variable {et_ctrl(5)}

# build widget .ets.frame0.frame17.radiobutton29
radiobutton .ets.frame0.frame17.radiobutton29 -text {File} -value {FILE} -variable {et_ctrl(5)}

# build widget .ets.frame0.frame18
frame .ets.frame0.frame18 -height {30} -width {30}

# build widget .ets.frame0.frame18.label11
label .ets.frame0.frame18.label11 -background {darkseagreen} -relief {raised} -text {E6} -width {9}

# build widget .ets.frame0.frame18.label13
label .ets.frame0.frame18.label13 -background {darkseagreen} -relief {raised} -textvariable {et_desc(6)}

# build widget .ets.frame0.frame18.label17
label .ets.frame0.frame18.label17 -background {mediumaquamarine} -relief {raised} -text {0} -textvariable {et(6)} -width {8}

# build widget .ets.frame0.frame18.entry13
entry .ets.frame0.frame18.entry13 -background {mediumturquoise} -font {-adobe-helvetica-medium-o-normal-12-120-75-75-p-67-iso8859-1} -relief {raised} -textvariable {et_frame(6)} -width {14}

# build widget .ets.frame0.frame18.radiobutton27
radiobutton .ets.frame0.frame18.radiobutton27 -relief {raised} -text {ON} -value {ON} -variable {et_ctrl(6)}

# build widget .ets.frame0.frame18.radiobutton28
radiobutton .ets.frame0.frame18.radiobutton28 -relief {raised} -text {OFF} -value {OFF} -variable {et_ctrl(6)}

# build widget .ets.frame0.frame18.radiobutton29
radiobutton .ets.frame0.frame18.radiobutton29 -relief {raised} -text {File} -value {FILE} -variable {et_ctrl(6)}

# build widget .ets.frame0.frame19
frame .ets.frame0.frame19 -height {30} -width {30}

# build widget .ets.frame0.frame19.label11
label .ets.frame0.frame19.label11 -background {darkseagreen} -relief {raised} -text {E7} -width {9}

# build widget .ets.frame0.frame19.label13
label .ets.frame0.frame19.label13 -background {darkseagreen} -relief {raised} -textvariable {et_desc(7)}

# build widget .ets.frame0.frame19.label17
label .ets.frame0.frame19.label17 -background {mediumaquamarine} -relief {raised} -text {0} -textvariable {et(7)} -width {8}

# build widget .ets.frame0.frame19.entry13
entry .ets.frame0.frame19.entry13 -background {mediumturquoise} -font {-adobe-helvetica-medium-o-normal-12-120-75-75-p-67-iso8859-1} -relief {raised} -textvariable {et_frame(7)} -width {14}

# build widget .ets.frame0.frame19.radiobutton27
radiobutton .ets.frame0.frame19.radiobutton27 -relief {raised} -text {ON} -value {ON} -variable {et_ctrl(7)}

# build widget .ets.frame0.frame19.radiobutton28
radiobutton .ets.frame0.frame19.radiobutton28 -relief {raised} -text {OFF} -value {OFF} -variable {et_ctrl(7)}

# build widget .ets.frame0.frame19.radiobutton29
radiobutton .ets.frame0.frame19.radiobutton29 -relief {raised} -text {File} -value {FILE} -variable {et_ctrl(7)}

# build widget .ets.frame0.frame20
frame .ets.frame0.frame20 -height {30} -width {30}

# build widget .ets.frame0.frame20.label11
label .ets.frame0.frame20.label11 -background {darkseagreen} -relief {raised} -text {E8} -width {9}

# build widget .ets.frame0.frame20.label13
label .ets.frame0.frame20.label13 -background {darkseagreen} -relief {raised} -textvariable {et_desc(8)}

# build widget .ets.frame0.frame20.label17
label .ets.frame0.frame20.label17 -background {mediumaquamarine} -relief {raised} -text {0} -textvariable {et(8)} -width {8}

# build widget .ets.frame0.frame20.entry13
entry .ets.frame0.frame20.entry13 -background {mediumturquoise} -font {-adobe-helvetica-medium-o-normal-12-120-75-75-p-67-iso8859-1} -relief {raised} -textvariable {et_frame(8)} -width {14}

# build widget .ets.frame0.frame20.radiobutton27
radiobutton .ets.frame0.frame20.radiobutton27 -relief {raised} -text {ON} -value {ON} -variable {et_ctrl(8)}
```

```

# build widget .ets.frame0.frame20.radiobutton28
radiobutton .ets.frame0.frame20.radiobutton28 -relief {raised} -text {OFF} -value {OFF} -variable {et_ctrl(8)}

# build widget .ets.frame0.frame20.radiobutton29
radiobutton .ets.frame0.frame20.radiobutton29 -relief {raised} -text {File} -value {FILE} -variable {et_ctrl(8)}

# build widget .ets.frame0.frame21
frame .ets.frame0.frame21 -height {30} -width {30}

# build widget .ets.frame0.frame21.label11
label .ets.frame0.frame21.label11 -background {darkseagreen} -relief {raised} -text {E9} -width {9}

# build widget .ets.frame0.frame21.label13
label .ets.frame0.frame21.label13 -background {darkseagreen} -relief {raised} -textvariable {et_desc(9)}

# build widget .ets.frame0.frame21.label17
label .ets.frame0.frame21.label17 -background {mediumaquamarine} -relief {raised} -text {0} -textvariable {et(9)} -width {8}

# build widget .ets.frame0.frame21.entry13
entry .ets.frame0.frame21.entry13 -background {mediumturquoise} -font {-adobe-helvetica-medium-o-normal-12-120-75-75-p-67-iso8859-1} -relief {raised} -textvariable {et_fname(9)} -width {14}

# build widget .ets.frame0.frame21.radiobutton27
radiobutton .ets.frame0.frame21.radiobutton27 -relief {raised} -text {ON} -value {ON} -variable {et_ctrl(9)}

# build widget .ets.frame0.frame21.radiobutton28
radiobutton .ets.frame0.frame21.radiobutton28 -relief {raised} -text {OFF} -value {OFF} -variable {et_ctrl(9)}

# build widget .ets.frame0.frame21.radiobutton29
radiobutton .ets.frame0.frame21.radiobutton29 -relief {raised} -text {File} -value {FILE} -variable {et_ctrl(9)}

# build widget .ets.frame0.frame22
frame .ets.frame0.frame22 -height {30} -width {30}

# build widget .ets.frame0.frame22.label11
label .ets.frame0.frame22.label11 -background {darkseagreen} -relief {raised} -text {E10} -width {9}

# build widget .ets.frame0.frame22.label13
label .ets.frame0.frame22.label13 -background {darkseagreen} -relief {raised} -textvariable {et_desc(10)}

# build widget .ets.frame0.frame22.label17
label .ets.frame0.frame22.label17 -background {mediumaquamarine} -relief {raised} -text {0} -textvariable {et(10)} -width {8}

# build widget .ets.frame0.frame22.entry13
entry .ets.frame0.frame22.entry13 -background {mediumturquoise} -font {-adobe-helvetica-medium-o-normal-12-120-75-75-p-67-iso8859-1} -relief {raised} -textvariable {et_fname(10)} -width {14}

# build widget .ets.frame0.frame22.radiobutton27
radiobutton .ets.frame0.frame22.radiobutton27 -relief {raised} -text {ON} -value {ON} -variable {et_ctrl(10)}

# build widget .ets.frame0.frame22.radiobutton28
radiobutton .ets.frame0.frame22.radiobutton28 -relief {raised} -text {OFF} -value {OFF} -variable {et_ctrl(10)}

# build widget .ets.frame0.frame22.radiobutton29
radiobutton .ets.frame0.frame22.radiobutton29 -relief {raised} -text {File} -value {FILE} -variable {et_ctrl(10)}

# pack master .ets.frame10
pack configure .ets.frame10.label11 -side left
pack configure .ets.frame10.label17 -side left
pack configure .ets.frame10.label23 -side left
pack configure .ets.frame10.label12 -side left
pack configure .ets.frame10.label13 -expand 1 -fill x -side left

# pack master .ets.frame14
pack configure .ets.frame14.button18 -expand 1 -fill x -side left

# pack master .ets.frame0
pack configure .ets.frame0.frame10 -fill x
pack configure .ets.frame0.frame14 -fill x
pack configure .ets.frame0.frame15 -fill x
pack configure .ets.frame0.frame16 -fill x
pack configure .ets.frame0.frame17 -fill x
pack configure .ets.frame0.frame18 -fill x
pack configure .ets.frame0.frame19 -fill x

```



```

# pack master .ets.frame0.frame21
pack configure .ets.frame0.frame21.label11 -side left
pack configure .ets.frame0.frame21.label17 -side left
pack configure .ets.frame0.frame21.radiobutton27 -side left
pack configure .ets.frame0.frame21.radiobutton28 -side left
pack configure .ets.frame0.frame21.radiobutton29 -side left
pack configure .ets.frame0.frame21.entry13 -side left
pack configure .ets.frame0.frame21.label13 -expand 1 -fill x -side left

# pack master .ets.frame0.frame22
pack configure .ets.frame0.frame22.label11 -side left
pack configure .ets.frame0.frame22.label17 -side left
pack configure .ets.frame0.frame22.radiobutton27 -side left
pack configure .ets.frame0.frame22.radiobutton28 -side left
pack configure .ets.frame0.frame22.radiobutton29 -side left
pack configure .ets.frame0.frame22.entry13 -side left
pack configure .ets.frame0.frame22.label13 -expand 1 -fill x -side left

# pack master .ets
pack configure .ets.frame10 -fill x
pack configure .ets.frame0 -fill x
pack configure .ets.frame14 -fill x

if {[info procs XFEdit] != ""} {
  catch "XFMiscBindWidgetTree .ets"
  after 2 "catch {XFEditSetShowWindows}"
}
}

proc DestroyWindow.ets {} {# xf ignore me 7
if {[info procs XFEdit] != ""} {
  if {[info commands .ets] != ""} {
    global xfShowWindow.ets
    set xfShowWindow.ets 0
    XFEditSetPath .
    after 2 "XFSaveAsProc .ets; XFEditSetShowWindows"
  }
} {
  catch "destroy .ets"
  update
}
}

# procedure to show window ShowWindow.ews
proc ShowWindow.ews { args } {
# xf ignore me 7

# build widget .ews
if {[info procs XFEdit] != ""} {
  catch "XFDestroy .ews"
} {
  catch "destroy .ews"
}
}
toplevel .ews -relief {raised}

# Window manager configurations
wm positionfrom .ews program
wm sizefrom .ews program
wm maxsize .ews 1000 864
wm minsize .ews 166 166
wm protocol .ews WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .ews {Event Summary}

# build widget .ews.frame
frame .ews.frame -relief {raised}

# build widget .ews.frame.scrollbar1
scrollbar .ews.frame.scrollbar1 -borderwidth {1} -command {.ews.frame.text2 yview} -highlightthickness {0} -relief {raised}

# build widget .ews.frame.text2
text .ews.frame.text2 -background {#c8dcdc} -borderwidth {1} -height {6} -width {40} -wrap {none} -yscrollcommand {.ews.frame.scrollbar1 set}

```



```

# build widget .evs.frame32
frame .evs.frame32 -height {30} -width {30}

# build widget .evs.frame32.button34
button .evs.frame32.button34 -borderwidth {1} -command {DestroyWindow.evs} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11}
-pady {4} -state {active} -text {Done}

# build widget .evs.frame32.button17
button .evs.frame32.button17 -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -text {Print}

# build widget .evs.frame32.button19
button .evs.frame32.button19 -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -text {Save}

# pack master .evs.frame
pack configure .evs.frame.scrollbar1 -fill y -side right
pack configure .evs.frame.text2 -expand 1 -fill both

# pack master .evs.frame32
pack configure .evs.frame32.button19 -side right
pack configure .evs.frame32.button17 -side right
pack configure .evs.frame32.button34 -side right

# pack master .evs
pack configure .evs.frame -expand 1 -fill both
pack configure .evs.frame32 -anchor e -side bottom

.evs.frame.text2 insert end {}

if {"[info procs XFEedit]" != ""} {
  catch "XFMiscBindWidgetTree .evs"
  after 2 "catch {XFEeditSetShowWindows}"
}
}

proc DestroyWindow.evs {} {# xf ignore me 7
if {"[info procs XFEedit]" != ""} {
  if {"[info commands .evs]" != ""} {
    global xfShowWindow.evs
    set xfShowWindow.evs 0
    XFEeditSetPath .
    after 2 "XFSaveAsProc .evs; XFEeditSetShowWindows"
  }
} {
  catch "destroy .evs"
  update
}
}

# procedure to show window ShowWindow.icsum
proc ShowWindow.icsum { args } {
# xf ignore me 7

# build widget .icsum
if {"[info procs XFEedit]" != ""} {
  catch "XFDestroy .icsum"
} {
  catch "destroy .icsum"
}
}
toplevel .icsum -relief {raised}

# Window manager configurations
wm positionfrom .icsum program
wm sizefrom .icsum program
wm maxsize .icsum 1152 864
wm minsize .icsum 570 283
wm protocol .icsum WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .icsum {Initial Conditions Summary}

# build widget .icsum.frame

```

```

frame .icsum.frame

# build widget .icsum.frame.scrollbar2
scrollbar .icsum.frame.scrollbar2 -borderwidth {1} -command {icsum.frame.listbox1 yview} -highlightthickness {0} -relief {raised}

# build widget .icsum.frame.listbox1
listbox .icsum.frame.listbox1 -background {grey90} -borderwidth {1} -font {-adobe-courier-bold-o-normal--14-140-75-75-m-90-iso8859-1} -scrollcommand
{icsum.frame.scrollbar2 set}
# bindings
bind .icsum.frame.listbox1 <ButtonRelease-1> {set index [%W nearest %y]
%W selection set $index
set line [%W get $index]
set icselected [lindex $line 0]
set icnewdesc [lrange $line 4 end]
if {[string range $icnewdesc 0 0] == ""} {
    set icnewdesc [string range $icnewdesc 1 end]
}}
bind .icsum.frame.listbox1 <Map> {read_ics}

# build widget .icsum.frame.frame0
frame .icsum.frame.frame0 -height {30} -width {30}

# build widget .icsum.frame.frame0.label1
label .icsum.frame.frame0.label1 -background {lightsteelblue4} -borderwidth {1} -foreground {white} -relief {raised} -text {IC No.} -width {6}

# build widget .icsum.frame.frame0.label2
label .icsum.frame.frame0.label2 -background {lightsteelblue4} -borderwidth {1} -foreground {white} -relief {raised} -text {Date} -width {12}

# build widget .icsum.frame.frame0.label3
label .icsum.frame.frame0.label3 -background {lightsteelblue4} -borderwidth {1} -foreground {white} -relief {raised} -text {Time} -width {12}

# build widget .icsum.frame.frame0.label4
label .icsum.frame.frame0.label4 -background {lightsteelblue4} -borderwidth {1} -foreground {white} -relief {raised} -text {Load Name} -width {17}

# build widget .icsum.frame.frame0.label5
label .icsum.frame.frame0.label5 -background {lightsteelblue4} -borderwidth {1} -foreground {white} -relief {raised} -text {Description}

# build widget .icsum.frame1
frame .icsum.frame1 -height {30} -width {30}

# build widget .icsum.frame1.button2
button .icsum.frame1.button2 -borderwidth {1} -command {css_cmd recd "reset IC$icselected"} -font {-adobe-helvetica-medium-r-normal--14-140-75-p-77-iso8859-1}
-padx {11} -pady {4} -text {Reset to}

# build widget .icsum.frame1.button3
button .icsum.frame1.button3 -borderwidth {1} -command {icsum_snap} -font {-adobe-helvetica-medium-r-normal--14-140-75-p-77-iso8859-1} -padx {11} -pady {4}
-text {Snap to}

# build widget .icsum.frame1.button4
button .icsum.frame1.button4 -borderwidth {1} -command {DestroyWindow.icsum} -font {-adobe-helvetica-medium-r-normal--14-140-75-p-77-iso8859-1} -padx {11}
-pady {4} -state {active} -text {Done}

# build widget .icsum.frame0
frame .icsum.frame0 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .icsum.frame0.label1
label .icsum.frame0.label1 -background {lightsteelblue4} -borderwidth {1} -foreground {white} -relief {raised} -text {Change description for IC:}

# build widget .icsum.frame0.label2
label .icsum.frame0.label2 -background {lightsteelblue2} -borderwidth {1} -foreground {red} -relief {raised} -text {50} -textvariable {icselected}

# build widget .icsum.frame0.entry3
entry .icsum.frame0.entry3 -background {azure} -borderwidth {1} -highlightthickness {0} -textvariable {icnewdesc}
# bindings
bind .icsum.frame0.entry3 <Key-Return> {icsum_desc}

# pack master .icsum.frame
pack configure .icsum.frame.scrollbar2 -fill y -side right
pack configure .icsum.frame.frame0 -fill x
pack configure .icsum.frame.listbox1 -expand 1 -fill both

# pack master .icsum.frame.frame0
pack configure .icsum.frame.frame0.label1 -side left
pack configure .icsum.frame.frame0.label2 -side left
pack configure .icsum.frame.frame0.label3 -side left

```



```

.icsum.frame.listbox1 insert end { 58 01/01/97 08:00:00 rtsched Description for IC 58}
.icsum.frame.listbox1 insert end { 59 01/01/97 08:00:00 rtsched Description for IC 59}
.icsum.frame.listbox1 insert end { 60 01/01/97 08:00:00 rtsched Description for IC 60}
.icsum.frame.listbox1 insert end { 61 01/01/97 08:00:00 rtsched Description for IC 61}
.icsum.frame.listbox1 insert end { 62 01/01/97 08:00:00 rtsched Description for IC 62}
.icsum.frame.listbox1 insert end { 63 01/01/97 08:00:00 rtsched Description for IC 63}
.icsum.frame.listbox1 insert end { 64 01/01/97 08:00:00 rtsched Description for IC 64}
.icsum.frame.listbox1 insert end { 65 01/01/97 08:00:00 rtsched Description for IC 65}
.icsum.frame.listbox1 insert end { 66 01/01/97 08:00:00 rtsched Description for IC 66}
.icsum.frame.listbox1 insert end { 67 01/01/97 08:00:00 rtsched Description for IC 67}
.icsum.frame.listbox1 insert end { 68 01/01/97 08:00:00 rtsched Description for IC 68}
.icsum.frame.listbox1 insert end { 69 01/01/97 08:00:00 rtsched Description for IC 69}
.icsum.frame.listbox1 insert end { 70 01/01/97 08:00:00 rtsched Description for IC 70}
.icsum.frame.listbox1 insert end { 71 01/01/97 08:00:00 rtsched Description for IC 71}
.icsum.frame.listbox1 insert end { 72 01/01/97 08:00:00 rtsched Description for IC 72}
.icsum.frame.listbox1 insert end { 73 01/01/97 08:00:00 rtsched Description for IC 73}
.icsum.frame.listbox1 insert end { 74 01/01/97 08:00:00 rtsched Description for IC 74}
.icsum.frame.listbox1 insert end { 75 01/01/97 08:00:00 rtsched Description for IC 75}
.icsum.frame.listbox1 insert end { 76 01/01/97 08:00:00 rtsched Description for IC 76}
.icsum.frame.listbox1 insert end { 77 01/01/97 08:00:00 rtsched Description for IC 77}
.icsum.frame.listbox1 insert end { 78 01/01/97 08:00:00 rtsched Description for IC 78}
.icsum.frame.listbox1 insert end { 79 01/01/97 08:00:00 rtsched Description for IC 79}
.icsum.frame.listbox1 insert end { 80 01/01/97 08:00:00 rtsched Description for IC 80}
.icsum.frame.listbox1 insert end { 81 01/01/97 08:00:00 rtsched Description for IC 81}
.icsum.frame.listbox1 insert end { 82 01/01/97 08:00:00 rtsched Description for IC 82}
.icsum.frame.listbox1 insert end { 83 01/01/97 08:00:00 rtsched Description for IC 83}
.icsum.frame.listbox1 insert end { 84 01/01/97 08:00:00 rtsched Description for IC 84}
.icsum.frame.listbox1 insert end { 85 01/01/97 08:00:00 rtsched Description for IC 85}
.icsum.frame.listbox1 insert end { 86 01/01/97 08:00:00 rtsched Description for IC 86}
.icsum.frame.listbox1 insert end { 87 01/01/97 08:00:00 rtsched Description for IC 87}
.icsum.frame.listbox1 insert end { 88 01/01/97 08:00:00 rtsched Description for IC 88}
.icsum.frame.listbox1 insert end { 89 01/01/97 08:00:00 rtsched Description for IC 89}
.icsum.frame.listbox1 insert end { 90 01/01/97 08:00:00 rtsched Description for IC 90}
.icsum.frame.listbox1 insert end { 91 01/01/97 08:00:00 rtsched Description for IC 91}
.icsum.frame.listbox1 insert end { 92 01/01/97 08:00:00 rtsched Description for IC 92}
.icsum.frame.listbox1 insert end { 93 01/01/97 08:00:00 rtsched Description for IC 93}
.icsum.frame.listbox1 insert end { 94 01/01/97 08:00:00 rtsched Description for IC 94}
.icsum.frame.listbox1 insert end { 95 01/01/97 08:00:00 rtsched Description for IC 95}
.icsum.frame.listbox1 insert end { 96 01/01/97 08:00:00 rtsched Description for IC 96}
.icsum.frame.listbox1 insert end { 97 01/01/97 08:00:00 rtsched Description for IC 97}
.icsum.frame.listbox1 insert end { 98 01/01/97 08:00:00 rtsched Description for IC 98}
.icsum.frame.listbox1 insert end { 99 01/01/97 08:00:00 rtsched Description for IC 99}

```

```

if {"[info procs XFEdit]" != ""} {
  catch "XFMiscBindWidgetTree .icsum"
  after 2 "catch {XFEditSetShowWindows}"
}
}

```

```

proc DestroyWindow.icsum {} {# xf ignore me 7
if {"[info procs XFEdit]" != ""} {
  if {"[info commands .icsum]" != ""} {
    global xfShowWindow.icsum
    set xfShowWindow.icsum 0
    XFEditSetPath .
    after 2 "XFSaveAsProc .icsum; XFEditSetShowWindows"
  }
} {
  catch "destroy .icsum"
  update
}
}

```

```

# procedure to show window ShowWindow.mfi
proc ShowWindow.mfi { args } {
# xf ignore me 7

```

```

# build widget .mfi
if {"[info procs XFEdit]" != ""} {
  catch "XFDestroy .mfi"
} {
  catch "destroy .mfi"
}

```

```

}
toplevel .mfi -relief {raised}

# Window manager configurations
wm positionfrom .mfi program
wm sizefrom .mfi program
wm maxsize .mfi 1152 864
wm minsize .mfi 490 300
wm protocol .mfi WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .mfi {Malfunction Index}

# build widget .mfi.frame
frame .mfi.frame

# build widget .mfi.frame.scrollbar2
scrollbar .mfi.frame.scrollbar2 -borderwidth {1} -command {.mfi.frame.listbox1 yview} -highlightthickness {0} -relief {raised}

# build widget .mfi.frame.listbox1
listbox .mfi.frame.listbox1 -background {#c8dcdc} -borderwidth {1} -font {-adobe-courier-bold-r-normal--12-120-75-75-m-70-iso8859-1} -yscrollcommand
{.mfi.frame.scrollbar2 set}
# bindings
bind .mfi.frame.listbox1 <Double-Button-1> {set index [.mfi.frame.listbox1 nearest %y]}
if {$index != ""} {
    mfp [lindex [.mfi.frame.listbox1 get $index] 0]
}
}

# build widget .mfi.frame.frame30
frame .mfi.frame.frame30 -height {30} -width {30}

# build widget .mfi.frame.frame30.label31
label .mfi.frame.frame30.label31 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -foreground {white} -relief
{raised} -text {MF Name} -width {19}

# build widget .mfi.frame.frame30.label33
label .mfi.frame.frame30.label33 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -foreground {white} -relief
{raised} -text {Description} -width {21}

# build widget .mfi.frame35
frame .mfi.frame35 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .mfi.frame35.label38
label .mfi.frame35.label38 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {white} -relief
{raised} -text {Search: }

# build widget .mfi.frame35.entry39
entry .mfi.frame35.entry39 -background {lightblue} -borderwidth {1} -font {-adobe-helvetica-medium-o-normal--14-140-75-75-p-78-iso8859-1} -highlightthickness {0}
-width {10}
# bindings
bind .mfi.frame35.entry39 <Key-Return> {mfi_fill [%W get]}

# build widget .mfi.frame21
frame .mfi.frame21 -height {30} -width {30}

# build widget .mfi.frame21.button36
button .mfi.frame21.button36 -borderwidth {1} -command {set index [.mfi.frame.listbox1 curselection]}
if {$index != ""} {
    mfp [lindex [.mfi.frame.listbox1 get $index] 0]
}
} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -text {Edit}

# build widget .mfi.frame21.button0
button .mfi.frame21.button0 -borderwidth {1} -command {ShowWindow.mfs} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4}
-text {Summary}

# build widget .mfi.frame21.button37
button .mfi.frame21.button37 -borderwidth {1} -command {.mfi.frame35.entry39 delete 0 end
DestroyWindow.mfi} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -state {active} -text {Done}

# pack master .mfi.frame
pack configure .mfi.frame.scrollbar2 -fill y -side right
pack configure .mfi.frame.frame30 -fill x
pack configure .mfi.frame.listbox1 -expand 1 -fill both

# pack master .mfi.frame.frame30

```

```

pack configure .mfi.frame.frame30.label31 -side left
pack configure .mfi.frame.frame30.label33 -expand 1 -fill x -side left

# pack master .mfi.frame35
pack configure .mfi.frame35.label38 -fill y -side left
pack configure .mfi.frame35.entry39 -expand 1 -fill both -side left

# pack master .mfi.frame21
pack configure .mfi.frame21.button37 -expand 1 -fill x -side left
pack configure .mfi.frame21.button36 -expand 1 -fill x -side left
pack configure .mfi.frame21.button0 -expand 1 -fill x -side left

# pack master .mfi
pack configure .mfi.frame -expand 1 -fill both
pack configure .mfi.frame35 -fill x
pack configure .mfi.frame21 -anchor e -side bottom

.mfi.frame.listbox1 insert end {edvreg_fail Voltage regulator setpt failure}
.mfi.frame.listbox1 insert end {edvreg_loss Loss of voltage regulator}
.mfi.frame35.entry39 insert end {}

if {[info procs XFEdit] != ""} {
    catch "XFMiscBindWidgetTree .mfi"
    after 2 "catch {XFEditSetShowWindows}"
}
}

proc DestroyWindow.mfi {} {# xf ignore me 7
if {[info procs XFEdit] != ""} {
    if {[info commands .mfi] != ""} {
        global xfShowWindow.mfi
        set xfShowWindow.mfi 0
        XFEditSetPath .
        after 2 "XFSaveAsProc .mfi; XFEditSetShowWindows"
    }
}
}
catch "destroy .mfi"
update
}
}

# procedure to show window ShowWindow.mfp
proc ShowWindow.mfp { args } {
# xf ignore me 7

# build widget .mfp
if {[info procs XFEdit] != ""} {
    catch "XFDestroy .mfp"
}
}
catch "destroy .mfp"
}
toplevel .mfp -relief {raised}

# Window manager configurations
wm positionfrom .mfp program
wm sizefrom .mfp program
wm maxsize .mfp 1000 768
wm minsize .mfp 104 10
wm protocol .mfp WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .mfp {Malfunction Control}

# build widget .mfp.frame16
frame .mfp.frame16 -height {30} -width {30}

# build widget .mfp.frame16.label24
label .mfp.frame16.label24 -borderwidth {0} -text {Initial severity} -width {12}

# build widget .mfp.frame16.entry28
entry .mfp.frame16.entry28 -background {azure} -borderwidth {1} -textvariable {init_severity_req}

```

```

# build widget .mfp.frame17
frame .mfp.frame17 -height {30} -width {30}

# build widget .mfp.frame17.label25
label .mfp.frame17.label25 -borderwidth {0} -text {Ramp time} -width {12}

# build widget .mfp.frame17.entry30
entry .mfp.frame17.entry30 -background {azure} -borderwidth {1} -textvariable {ramp_time_req}

# build widget .mfp.frame18
frame .mfp.frame18 -height {30} -width {30}

# build widget .mfp.frame18.label26
label .mfp.frame18.label26 -borderwidth {0} -text {Final severity} -width {12}

# build widget .mfp.frame18.entry31
entry .mfp.frame18.entry31 -background {azure} -borderwidth {1} -textvariable {final_severity_req}

# build widget .mfp.frame19
frame .mfp.frame19 -height {30} -width {30}

# build widget .mfp.frame19.label27
label .mfp.frame19.label27 -borderwidth {0} -text {Delay} -width {12}

# build widget .mfp.frame19.entry32
entry .mfp.frame19.entry32 -background {azure} -borderwidth {1} -textvariable {delay_req}

# build widget .mfp.frame20
frame .mfp.frame20 -height {30} -width {30}

# build widget .mfp.frame20.button21
button .mfp.frame20.button21 -borderwidth {1} -command {mfp_apply} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4}
-text {Apply}

# build widget .mfp.frame20.button23
button .mfp.frame20.button23 -borderwidth {1} -command {DestroyWindow.mfp} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11}
-pady {4} -state {active} -text {Done}

# build widget .mfp.frame33
frame .mfp.frame33 -height {30} -width {30}

# build widget .mfp.frame33.label25
label .mfp.frame33.label25 -borderwidth {0} -text {Trigger} -width {12}

# build widget .mfp.frame33.entry30
entry .mfp.frame33.entry30 -background {azure} -borderwidth {1} -textvariable {trigger_req}

# build widget .mfp.frame34
frame .mfp.frame34 -borderwidth {4} -height {30} -width {30}

# build widget .mfp.frame34.label35
label .mfp.frame34.label35 -borderwidth {0} -text {Status}

# build widget .mfp.frame34.checkbutton37
checkbutton .mfp.frame34.checkbutton37 -background {steelblue} -foreground {green} -indicatoron {0} -offvalue {OFF} -onvalue {Active} -relief {raised} -text {OFF}
-textvariable {mfstatus} -variable {mfstatus} -width {6}

# build widget .mfp.frame2
frame .mfp.frame2 -borderwidth {2} -height {30} -relief {raised} -width {30}

# build widget .mfp.frame2.label15
label .mfp.frame2.label15 -borderwidth {0} -font {-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1} -foreground {red} -padx {8} -pady {2} -text {Voltage
regulator setpt failure} -textvariable {mfpdesc}

# build widget .mfp.frame2.label0
label .mfp.frame2.label0 -borderwidth {0} -font {-adobe-times-bold-i-normal--14-140-75-75-p-77-iso8859-1} -foreground {red} -text {edvreg_fail} -textvariable
{mfp_mfname}

# pack master .mfp.frame16
pack configure .mfp.frame16.label24 -side left
pack configure .mfp.frame16.entry28 -side left

# pack master .mfp.frame17
pack configure .mfp.frame17.label25 -side left
pack configure .mfp.frame17.entry30 -side left

```

```

# pack master .mfp.frame18
pack configure .mfp.frame18.label26 -side left
pack configure .mfp.frame18.entry31 -side left

# pack master .mfp.frame19
pack configure .mfp.frame19.label27 -side left
pack configure .mfp.frame19.entry32 -side left

# pack master .mfp.frame20
pack configure .mfp.frame20.button23 -expand 1 -fill x -side left
pack configure .mfp.frame20.button21 -expand 1 -fill x -side left

# pack master .mfp.frame33
pack configure .mfp.frame33.label25 -side left
pack configure .mfp.frame33.entry30 -side left

# pack master .mfp.frame34
pack configure .mfp.frame34.label35 -side left
pack configure .mfp.frame34.checkbutton37

# pack master .mfp.frame2
pack configure .mfp.frame2.label15 -fill x
pack configure .mfp.frame2.label0

# pack master .mfp
pack configure .mfp.frame2 -fill x
pack configure .mfp.frame34
pack configure .mfp.frame16
pack configure .mfp.frame17
pack configure .mfp.frame18
pack configure .mfp.frame19
pack configure .mfp.frame33
pack configure .mfp.frame20 -fill x

if {[info procs XFEdit] != ""} {
  catch "XFMiscBindWidgetTree .mfp"
  after 2 "catch {XFEditSetShowWindows}"
}
}

proc DestroyWindow.mfp {} {# xf ignore me 7
if {[info procs XFEdit] != ""} {
  if {[info commands .mfp] != ""} {
    global xfShowWindow.mfp
    set xfShowWindow.mfp 0
    XFEditSetPath .
    after 2 "XFSaveAsProc .mfp; XFEditSetShowWindows"
  }
} {
  catch "destroy .mfp"
  update
}
}

# procedure to show window ShowWindow.mfs
proc ShowWindow.mfs { args } {
# xf ignore me 7

# build widget .mfs
if {[info procs XFEdit] != ""} {
  catch "XFDestroy .mfs"
} {
  catch "destroy .mfs"
}
toplevel .mfs -relief {raised}

# Window manager configurations
wm positionfrom .mfs program
wm sizefrom .mfs program
wm iconname .mfs {calvin.ico}
wm maxsize .mfs 1000 864
wm minsize .mfs 300 300
wm protocol .mfs WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.

```


Please use the "Current widget path:" to show/hide windows.}}
wm title .mfs {Malfunction summary}

```
# build widget .mfs.frame  
frame .mfs.frame
```

```
# build widget .mfs.frame.scrollbar2  
scrollbar .mfs.frame.scrollbar2 -borderwidth {1} -command {mfs.frame.listbox1 yview} -highlightthickness {0} -relief {raised}
```

```
# build widget .mfs.frame.listbox1  
listbox .mfs.frame.listbox1 -background {#c8dcdc} -borderwidth {1} -font {-adobe-courier-bold-r-normal-14-140-75-75-m-90-iso8859-1} -yscrollcommand  
{mfs.frame.scrollbar2 set}
```

```
# build widget .mfs.frame.frame30  
frame .mfs.frame.frame30 -height {30} -width {30}
```

```
# build widget .mfs.frame.frame30.label31  
label .mfs.frame.frame30.label31 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal-12-120-75-75-p-70-iso8859-1} -foreground {white} -relief  
{raised} -text {Malfunction} -width {20}
```

```
# build widget .mfs.frame.frame30.label33  
label .mfs.frame.frame30.label33 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal-12-120-75-75-p-70-iso8859-1} -foreground {white} -relief  
{raised} -text {Description} -width {21}
```

```
# build widget .mfs.frame.frame30.label34  
label .mfs.frame.frame30.label34 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal-12-120-75-75-p-70-iso8859-1} -foreground {white} -relief  
{raised} -text {Status} -width {26}
```

```
# build widget .mfs.frame35  
frame .mfs.frame35 -height {30} -width {30}
```

```
# build widget .mfs.frame35.button36  
button .mfs.frame35.button36 -borderwidth {1} -command {set index [mfs.frame.listbox1 curselection]  
if {$index != ""}{  
    mfp [lindex [mfs.frame.listbox1 get $index] 0]  
}} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -text {Edit}
```

```
# build widget .mfs.frame35.button37  
button .mfs.frame35.button37 -borderwidth {1} -command {DestroyWindow.mfs} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11}  
-pady {4} -state {active} -text {Done}
```

```
# build widget .mfs.frame35.button2  
button .mfs.frame35.button2 -borderwidth {1} -command {ShowWindow.mf} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4}  
-text {Index}
```

```
# pack master .mfs.frame  
pack configure .mfs.frame.scrollbar2 -fill y -side right  
pack configure .mfs.frame.frame30 -fill x  
pack configure .mfs.frame.listbox1 -expand 1 -fill both
```

```
# pack master .mfs.frame.frame30  
pack configure .mfs.frame.frame30.label31 -side left  
pack configure .mfs.frame.frame30.label34 -side left  
pack configure .mfs.frame.frame30.label33 -expand 1 -fill x -side left
```

```
# pack master .mfs.frame35  
pack configure .mfs.frame35.button37 -side left  
pack configure .mfs.frame35.button36 -side left  
pack configure .mfs.frame35.button2 -side left
```

```
# pack master .mfs  
pack configure .mfs.frame -expand 1 -fill both  
pack configure .mfs.frame35 -anchor e -side bottom
```

```
if {[info procs XFEdit] != ""}  
    catch "XFMiscBindWidgetTree .mfs"  
    after 2 "catch {XFEditSetShowWindows}"  
}
```

```
proc DestroyWindow.mfs {} {# xf ignore me 7  
    if {[info procs XFEdit] != ""}  
        if {[info commands .mfs] != ""}  
            global xShowWindow.mfs
```

```

set xfShowWindow.mfs 0
XFeditSelPath .
after 2 "XFSaveAsProc .mfs; XFeditSetShowWindows"
}
}
catch "destroy .mfs"
update
}
}

```

```

# procedure to show window ShowWindow.monparm
proc ShowWindow.monparm { args } {
# xf ignore me 7

```

```

# build widget .monparm
if {"[info procs XFedit]" != ""} {
catch "XFDestroy .monparm"
}
catch "destroy .monparm"
}
toplevel .monparm

```

```

# Window manager configurations
wm positionfrom .monparm program
wm sizefrom .monparm program
wm geometry .monparm 523x513+189+135
wm maxsize .monparm 523 690
wm minsize .monparm 249 167
wm protocol .monparm WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .monparm {Monitored Parameters}

```

```

# build widget .monparm.frame36
frame .monparm.frame36 -height {30} -width {30}

```

```

# build widget .monparm.frame36.button0
button .monparm.frame36.button0 -borderwidth {1} -command {DestroyWindow.monparm} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -state {active} -text {Done}

```

```

# build widget .monparm.frame36.button1
button .monparm.frame36.button1 -borderwidth {1} -command {ShowWindow.query} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -text {Query}

```

```

# build widget .monparm.frame36.button2
button .monparm.frame36.button2 -borderwidth {1} -command {mps_load} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -text {Load}

```

```

# build widget .monparm.frame36.button3
button .monparm.frame36.button3 -borderwidth {1} -command {mps_insert} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -text {Insert}

```

```

# build widget .monparm.frame36.button4
button .monparm.frame36.button4 -borderwidth {1} -command {mps_save} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -text {Save}

```

```

# build widget .monparm.frame37
frame .monparm.frame37 -borderwidth {2} -height {30} -relief {raised} -width {30}

```

```

# build widget .monparm.frame37.frame38
frame .monparm.frame37.frame38 -borderwidth {2} -height {30} -relief {ridge} -width {30}

```

```

# build widget .monparm.frame37.frame38.name0
label .monparm.frame37.frame38.name0 -background {grey70} -relief {raised} -textvariable {name(0)} -width {21}
# bindings
bind .monparm.frame37.frame38.name0 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name0 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name0 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name0 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name0 <Enter> {mps_catch %W}

```

```

# build widget .monparm.frame37.frame38.name1
label .monparm.frame37.frame38.name1 -background {grey80} -relief {raised} -text {edvoltreg} -textvariable {name(1)} -width {21}
# bindings

```

```
bind .monparm.frame37.frame38.name1 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name1 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name1 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name1 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name1 <Enter> {mps_catch %W}
```

```
# build widget .monparm.frame37.frame38.name2
```

```
label .monparm.frame37.frame38.name2 -background {grey70} -relief {raised} -textvariable {name(2)} -width {21}
```

```
# bindings
```

```
bind .monparm.frame37.frame38.name2 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name2 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name2 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name2 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name2 <Enter> {mps_catch %W}
```

```
# build widget .monparm.frame37.frame38.name3
```

```
label .monparm.frame37.frame38.name3 -background {grey80} -font {-adobe-helvetica-bold-o-normal--12-120-75-75-p-69-iso8859-1} -relief {raised} -text {edibattery}
-textvariable {name(3)} -width {21}
```

```
# bindings
```

```
bind .monparm.frame37.frame38.name3 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name3 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name3 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name3 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name3 <Enter> {mps_catch %W}
```

```
# build widget .monparm.frame37.frame38.name4
```

```
label .monparm.frame37.frame38.name4 -background {grey70} -relief {raised} -textvariable {name(4)} -width {21}
```

```
# bindings
```

```
bind .monparm.frame37.frame38.name4 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name4 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name4 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name4 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name4 <Enter> {mps_catch %W}
```

```
# build widget .monparm.frame37.frame38.name5
```

```
label .monparm.frame37.frame38.name5 -background {grey80} -relief {raised} -textvariable {name(5)} -width {21}
```

```
# bindings
```

```
bind .monparm.frame37.frame38.name5 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name5 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name5 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name5 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name5 <Enter> {mps_catch %W}
```

```
# build widget .monparm.frame37.frame38.name6
```

```
label .monparm.frame37.frame38.name6 -background {grey70} -relief {raised} -textvariable {name(6)} -width {21}
```

```
# bindings
```

```
bind .monparm.frame37.frame38.name6 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name6 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name6 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name6 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name6 <Enter> {mps_catch %W}
```

```
# build widget .monparm.frame37.frame38.name7
```

```
label .monparm.frame37.frame38.name7 -background {grey80} -relief {raised} -textvariable {name(7)} -width {21}
```

```
# bindings
```

```
bind .monparm.frame37.frame38.name7 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name7 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name7 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name7 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name7 <Enter> {mps_catch %W}
```

```
# build widget .monparm.frame37.frame38.name8
```

```
label .monparm.frame37.frame38.name8 -background {grey70} -relief {raised} -textvariable {name(8)} -width {21}
```

```
# bindings
```

```
bind .monparm.frame37.frame38.name8 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name8 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name8 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name8 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name8 <Enter> {mps_catch %W}
```

```
# build widget .monparm.frame37.frame38.name9
```

```
label .monparm.frame37.frame38.name9 -background {grey80} -relief {raised} -textvariable {name(9)} -width {21}
```

```
# bindings
```

```
bind .monparm.frame37.frame38.name9 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name9 <Button-1> {label_drag %W %X %Y}
```



```

# bindings
bind .monparm.frame37.frame38.name26 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name26 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name26 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name26 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name26 <Enter> {mps_catch %W}

# build widget .monparm.frame37.frame38.name27
label .monparm.frame37.frame38.name27 -background {grey80} -relief {raised} -textvariable {name(27)} -width {21}
# bindings
bind .monparm.frame37.frame38.name27 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name27 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name27 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name27 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name27 <Enter> {mps_catch %W}

# build widget .monparm.frame37.frame38.name28
label .monparm.frame37.frame38.name28 -background {grey70} -relief {raised} -textvariable {name(28)} -width {21}
# bindings
bind .monparm.frame37.frame38.name28 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name28 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name28 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name28 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name28 <Enter> {mps_catch %W}

# build widget .monparm.frame37.frame38.name29
label .monparm.frame37.frame38.name29 -background {grey80} -relief {raised} -textvariable {name(29)} -width {21}
# bindings
bind .monparm.frame37.frame38.name29 <B1-Motion> {move.dragwin %X %Y}
bind .monparm.frame37.frame38.name29 <Button-1> {label_drag %W %X %Y}
bind .monparm.frame37.frame38.name29 <Button-3> {label_setvar %W}
bind .monparm.frame37.frame38.name29 <ButtonRelease-1> {drop.dragwin}
bind .monparm.frame37.frame38.name29 <Enter> {mps_catch %W}

# build widget .monparm.frame37.frame38.mpscroll
scrollbar .monparm.frame37.frame38.mpscroll -activebackground {lightsteelblue} -activerelief {sunken} -background {lightsteelblue} -borderwidth {1} -command
{mps_scroll} -highlightthickness {0} -relief {raised}
# bindings
bind .monparm.frame37.frame38.mpscroll <Map> {set first [expr $mps_base * 1.0 / $mps_maxvar]
set last [expr ($mps_base + $mps_height) * 1.0 / $mps_maxvar]
.monparm.frame37.frame38.mpscroll set $first $last}

# build widget .monparm.frame37.frame39
frame .monparm.frame37.frame39 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .monparm.frame37.frame39.value0
label .monparm.frame37.frame39.value0 -background {grey70} -relief {raised} -textvariable {value(0)} -width {12}
# bindings
bind .monparm.frame37.frame39.value0 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value1
label .monparm.frame37.frame39.value1 -background {grey80} -relief {raised} -textvariable {value(1)} -width {12}
# bindings
bind .monparm.frame37.frame39.value1 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value2
label .monparm.frame37.frame39.value2 -background {grey70} -relief {raised} -textvariable {value(2)} -width {12}
# bindings
bind .monparm.frame37.frame39.value2 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value3
label .monparm.frame37.frame39.value3 -background {grey80} -relief {raised} -textvariable {value(3)} -width {12}
# bindings
bind .monparm.frame37.frame39.value3 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value4
label .monparm.frame37.frame39.value4 -background {grey70} -relief {raised} -textvariable {value(4)} -width {12}
# bindings
bind .monparm.frame37.frame39.value4 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value5
label .monparm.frame37.frame39.value5 -background {grey80} -relief {raised} -textvariable {value(5)} -width {12}
# bindings
bind .monparm.frame37.frame39.value5 <Button-3> {label_setval %W}

```



```

# build widget .monparm.frame37.frame39.value21
label .monparm.frame37.frame39.value21 -background {grey80} -relief {raised} -textvariable {value(21)} -width {12}
# bindings
bind .monparm.frame37.frame39.value21 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value22
label .monparm.frame37.frame39.value22 -background {grey70} -relief {raised} -textvariable {value(22)} -width {12}
# bindings
bind .monparm.frame37.frame39.value22 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value23
label .monparm.frame37.frame39.value23 -background {grey80} -relief {raised} -textvariable {value(23)} -width {12}
# bindings
bind .monparm.frame37.frame39.value23 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value24
label .monparm.frame37.frame39.value24 -background {grey70} -relief {raised} -textvariable {value(24)} -width {12}
# bindings
bind .monparm.frame37.frame39.value24 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value25
label .monparm.frame37.frame39.value25 -background {grey80} -relief {raised} -textvariable {value(25)} -width {12}
# bindings
bind .monparm.frame37.frame39.value25 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value26
label .monparm.frame37.frame39.value26 -background {grey70} -relief {raised} -textvariable {value(26)} -width {12}
# bindings
bind .monparm.frame37.frame39.value26 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value27
label .monparm.frame37.frame39.value27 -background {grey80} -relief {raised} -textvariable {value(27)} -width {12}
# bindings
bind .monparm.frame37.frame39.value27 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value28
label .monparm.frame37.frame39.value28 -background {grey70} -relief {raised} -textvariable {value(28)} -width {12}
# bindings
bind .monparm.frame37.frame39.value28 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame39.value29
label .monparm.frame37.frame39.value29 -background {grey80} -relief {raised} -textvariable {value(29)} -width {12}
# bindings
bind .monparm.frame37.frame39.value29 <Button-3> {label_setval %W}

# build widget .monparm.frame37.frame40
frame .monparm.frame37.frame40 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .monparm.frame37.frame40.units0
label .monparm.frame37.frame40.units0 -background {grey70} -relief {raised} -textvariable {units(0)} -width {8}

# build widget .monparm.frame37.frame40.units1
label .monparm.frame37.frame40.units1 -background {grey80} -relief {raised} -textvariable {units(1)} -width {8}

# build widget .monparm.frame37.frame40.units2
label .monparm.frame37.frame40.units2 -background {grey70} -relief {raised} -textvariable {units(2)} -width {8}

# build widget .monparm.frame37.frame40.units3
label .monparm.frame37.frame40.units3 -background {grey80} -relief {raised} -textvariable {units(3)} -width {8}

# build widget .monparm.frame37.frame40.units4
label .monparm.frame37.frame40.units4 -background {grey70} -relief {raised} -textvariable {units(4)} -width {8}

# build widget .monparm.frame37.frame40.units5
label .monparm.frame37.frame40.units5 -background {grey80} -relief {raised} -textvariable {units(5)} -width {8}

# build widget .monparm.frame37.frame40.units6
label .monparm.frame37.frame40.units6 -background {grey70} -relief {raised} -textvariable {units(6)} -width {8}

# build widget .monparm.frame37.frame40.units7
label .monparm.frame37.frame40.units7 -background {grey80} -relief {raised} -textvariable {units(7)} -width {8}

# build widget .monparm.frame37.frame40.units8
label .monparm.frame37.frame40.units8 -background {grey70} -relief {raised} -textvariable {units(8)} -width {8}

```



```
# build widget .monparm.frame37.frame41.desc28
label .monparm.frame37.frame41.desc28 -background {grey70} -relief {raised} -textvariable {desc(28)} -width {33}

# build widget .monparm.frame37.frame41.desc29
label .monparm.frame37.frame41.desc29 -background {grey80} -relief {raised} -textvariable {desc(29)} -width {33}

# build widget .monparm.frame42
frame .monparm.frame42 -borderwidth {2} -height {30} -relief {raised} -width {30}

# build widget .monparm.frame42.label3
label .monparm.frame42.label43 -background {lightsteelblue} -borderwidth {0} -font {-adobe-helvetica-medium-o-normal--18-180-75-75-p-98-iso8859-1} -foreground {red}
-text {Name}

# build widget .monparm.frame42.label1
label .monparm.frame42.label1 -background {lightsteelblue} -borderwidth {0} -font {-adobe-helvetica-medium-o-normal--18-180-75-75-p-98-iso8859-1} -foreground {red}
-text {Value}

# build widget .monparm.frame42.label3
label .monparm.frame42.label3 -background {lightsteelblue} -borderwidth {0} -font {-adobe-helvetica-medium-o-normal--18-180-75-75-p-98-iso8859-1} -foreground {red}
-text {Units}

# build widget .monparm.frame42.label5
label .monparm.frame42.label5 -background {lightsteelblue} -borderwidth {0} -font {-adobe-helvetica-medium-o-normal--18-180-75-75-p-98-iso8859-1} -foreground {red}
-text {Description}

# pack master .monparm.frame36
pack configure .monparm.frame36.button0 -side left
pack configure .monparm.frame36.button2 -side left
pack configure .monparm.frame36.button3 -side left
pack configure .monparm.frame36.button4 -side left
pack configure .monparm.frame36.button1 -side left

# pack master .monparm.frame37
pack configure .monparm.frame37.frame38 -fill y -side left
pack configure .monparm.frame37.frame39 -fill y -side left
pack configure .monparm.frame37.frame40 -fill y -side left
pack configure .monparm.frame37.frame41 -fill y -side left

# pack master .monparm.frame37.frame38
pack configure .monparm.frame37.frame38.mpscroll -fill y -side left
pack configure .monparm.frame37.frame38.name0
pack configure .monparm.frame37.frame38.name1
pack configure .monparm.frame37.frame38.name2
pack configure .monparm.frame37.frame38.name3
pack configure .monparm.frame37.frame38.name4
pack configure .monparm.frame37.frame38.name5
pack configure .monparm.frame37.frame38.name6
pack configure .monparm.frame37.frame38.name7
pack configure .monparm.frame37.frame38.name8
pack configure .monparm.frame37.frame38.name9
pack configure .monparm.frame37.frame38.name10
pack configure .monparm.frame37.frame38.name11
pack configure .monparm.frame37.frame38.name12
pack configure .monparm.frame37.frame38.name13
pack configure .monparm.frame37.frame38.name14
pack configure .monparm.frame37.frame38.name15
pack configure .monparm.frame37.frame38.name16
pack configure .monparm.frame37.frame38.name17
pack configure .monparm.frame37.frame38.name18
pack configure .monparm.frame37.frame38.name19
pack configure .monparm.frame37.frame38.name20
pack configure .monparm.frame37.frame38.name21
pack configure .monparm.frame37.frame38.name22
pack configure .monparm.frame37.frame38.name23
pack configure .monparm.frame37.frame38.name24
pack configure .monparm.frame37.frame38.name25
pack configure .monparm.frame37.frame38.name26
pack configure .monparm.frame37.frame38.name27
pack configure .monparm.frame37.frame38.name28
pack configure .monparm.frame37.frame38.name29

# pack master .monparm.frame37.frame39
pack configure .monparm.frame37.frame39.value0
pack configure .monparm.frame37.frame39.value1
pack configure .monparm.frame37.frame39.value2
```



```

pack configure .monparm.frame37.frame41.desc14
pack configure .monparm.frame37.frame41.desc15
pack configure .monparm.frame37.frame41.desc16
pack configure .monparm.frame37.frame41.desc17
pack configure .monparm.frame37.frame41.desc18
pack configure .monparm.frame37.frame41.desc19
pack configure .monparm.frame37.frame41.desc20
pack configure .monparm.frame37.frame41.desc21
pack configure .monparm.frame37.frame41.desc22
pack configure .monparm.frame37.frame41.desc23
pack configure .monparm.frame37.frame41.desc24
pack configure .monparm.frame37.frame41.desc25
pack configure .monparm.frame37.frame41.desc26
pack configure .monparm.frame37.frame41.desc27
pack configure .monparm.frame37.frame41.desc28
pack configure .monparm.frame37.frame41.desc29

```

```

# pack master .monparm.frame42
pack configure .monparm.frame42.label43 -fill x -ipadx 60 -side left
pack configure .monparm.frame42.label1 -fill x -ipadx 18 -side left
pack configure .monparm.frame42.label3 -fill x -ipadx 10 -side left
pack configure .monparm.frame42.label5 -expand 1 -fill x -side left

```

```

# pack master .monparm
pack configure .monparm.frame36 -anchor e
pack configure .monparm.frame42 -fill x
pack configure .monparm.frame37 -expand 1 -fill both

```

```

if {[info procs XFEdit] != ""} {
  catch "XFMiscBindWidget(Tree .monparm)"
  after 2 "catch {XFEditSetShowWindows}"
}
}

```

```

proc DestroyWindow.monparm {} {# xf ignore me 7
  if {[info procs XFEdit] != ""} {
    if {[info commands .monparm] != ""} {
      global xfShowWindow.monparm
      set xfShowWindow.monparm 0
      XFEditSetPath .
      after 2 "XFSaveAsProc .monparm; XFEditSetShowWindows"
    }
  }
  catch "destroy .monparm"
  update
}
}

```

```

# procedure to show window ShowWindow.pref
proc ShowWindow.pref { args } {
  # xf ignore me 7

```

```

# build widget .pref
if {[info procs XFEdit] != ""} {
  catch "XFDestroy .pref"
}
catch "destroy .pref"
}
toplevel .pref -relief {raised}

```

```

# Window manager configurations
wm positionfrom .pref program
wm sizefrom .pref program
wm geometry .pref 260x116+109+117
wm maxsize .pref 260 116
wm minsize .pref 260 116
wm protocol .pref WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .pref {Preferences}

```

```

# bindings
bind .pref <Reparent> {set ics_alarm_new $ics_alarm
set ics_confirm_new $ics_confirm
set ics_restrict_new $ics_restrict}

```

```

# build widget .pref.frame9
frame .pref.frame9 -height {30} -width {30}

# build widget .pref.frame9.button11
button .pref.frame9.button11 -borderwidth {1} -command {set ics_alarm $ics_alarm_new
set ics_confirm $ics_confirm_new
set ics_restrict $ics_restrict_new
if {$ics_restrict == "on"} {
    css_cmd stat "restrict [info hostname] 0"
} { css_cmd stat "restrict [info hostname] 1" }
DestroyWindow.pref} -font {-adobe-helvetica-medium-r-normal-12-120-75-75-p-67-iso8859-1} -padx {11} -pady {4} -state {active} -text {Done}

# build widget .pref.frame9.button0
button .pref.frame9.button0 -borderwidth {1} -command {DestroyWindow.pref} -font {-adobe-helvetica-medium-r-normal-12-120-75-75-p-67-iso8859-1} -padx {11} -pady
{4} -text {Cancel}

# build widget .pref.checkbutton0
checkbutton .pref.checkbutton0 -font {-adobe-helvetica-bold-r-normal-12-120-75-75-p-70-iso8859-1} -offvalue {off} -onvalue {on} -text {Confirmation on} -variable
{ics_confirm_new}

# build widget .pref.checkbutton1
checkbutton .pref.checkbutton1 -font {-adobe-helvetica-bold-r-normal-12-120-75-75-p-70-iso8859-1} -offvalue {off} -onvalue {on} -text {Restricted} -variable
{ics_restrict_new}

# build widget .pref.checkbutton2
checkbutton .pref.checkbutton2 -font {-adobe-helvetica-bold-r-normal-12-120-75-75-p-70-iso8859-1} -offvalue {off} -onvalue {on} -text {Alarm pop-up} -variable
{ics_alarm_new}

# pack master .pref.frame9
pack configure .pref.frame9.button0 -side right
pack configure .pref.frame9.button11 -side right

# pack master .pref
pack configure .pref.frame9 -anchor s -expand 1 -fill x -side bottom
pack configure .pref.checkbutton0 -anchor w
pack configure .pref.checkbutton1 -anchor w
pack configure .pref.checkbutton2 -anchor w

if {[info procs XFEdit] != ""} {
    catch "XFMiscBindWidgetTree .pref"
    after 2 "catch {XFEditSetShowWindows}"
}
}

proc DestroyWindow.pref {} {# xf ignore me 7
if {[info procs XFEdit] != ""} {
    if {[info commands .pref] != ""} {
        global xfShowWindow.pref
        set xfShowWindow.pref 0
        XFEditSetPath .
        after 2 "XFSaveAsProc .pref; XFEditSetShowWindows"
    }
} {
    catch "destroy .pref"
    update
}
}

# procedure to show window ShowWindow.query
proc ShowWindow.query { args } {
# xf ignore me 7

# build widget .query
if {[info procs XFEdit] != ""} {
    catch "XFDestroy .query"
} {
    catch "destroy .query"
}
}
toplevel .query

# Window manager configurations
wm positionfrom .query program
wm sizefrom .query program

```

```
wm maxsize .query 1000 864
wm minsize .query 374 300
wm protocol .query WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .query {Query Programmer's Data Base}
```

```
# build widget .query.frame
frame .query.frame
```

```
# build widget .query.frame.scrollbar3
scrollbar .query.frame.scrollbar3 -borderwidth {1} -command {query.frame.listbox1 xview} -highlightthickness {0} -orient {horizontal} -relief {raised}
```

```
# build widget .query.frame.scrollbar2
scrollbar .query.frame.scrollbar2 -borderwidth {1} -command {query.frame.listbox1 yview} -highlightthickness {0} -relief {raised}
```

```
# build widget .query.frame.listbox1
listbox .query.frame.listbox1 -background {lightblue} -borderwidth {1} -font {-adobe-courier-bold-o-normal--12-120-75-75-m-70-iso8859-1} -xscrollcommand
{query.frame.scrollbar3 set} -yscrollcommand {query.frame.scrollbar2 set}
```

```
# bindings
```

```
bind .query.frame.listbox1 <B1-Motion> {%W select anchor [%W nearest %y]}
bind .query.frame.listbox1 <Button-1> {list_drag %W %y %X %Y}
bind .query.frame.listbox1 <ButtonRelease-1> {drop.dragwin}
bind .query.frame.listbox1 <Shift-B1-Motion> {%W select anchor [%W nearest %y]}
bind .query.frame.listbox1 <Shift-Button-1> {%W select anchor [%W nearest %y]}
```

```
# build widget .query.frame3
frame .query.frame3 -height {30} -width {30}
```

```
# build widget .query.frame3.button4
button .query.frame3.button4 -borderwidth {1} -command {DestroyWindow.query} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -padx {11}
-pady {4} -state {active} -text {Done}
```

```
# build widget .query.frame10
frame .query.frame10 -borderwidth {2} -height {30} -relief {ridge} -width {30}
```

```
# build widget .query.frame10.frame2
frame .query.frame10.frame2 -height {30} -width {30}
```

```
# build widget .query.frame10.frame2.label5
label .query.frame10.frame2.label5 -borderwidth {0} -font {-adobe-helvetica-bold-o-normal--12-120-75-75-p-69-iso8859-1} -text { Match: }
```

```
# build widget .query.frame10.frame2.entry6
entry .query.frame10.frame2.entry6 -background {lightblue} -font {-adobe-helvetica-bold-o-normal--12-120-75-75-p-69-iso8859-1} -textvariable {query_match} -width {16}
# bindings
bind .query.frame10.frame2.entry6 <Key-Return> {query_list $query_match}
```

```
# build widget .query.frame10.frame2.radiobutton12
radiobutton .query.frame10.frame2.radiobutton12 -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -highlightthickness {0} -text {items starting with}
-value {start} -variable {query_type}
```

```
# build widget .query.frame10.frame2.radiobutton13
radiobutton .query.frame10.frame2.radiobutton13 -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -highlightthickness {0} -text {items containing}
-value {in} -variable {query_type}
```

```
# build widget .query.frame10.frame4
frame .query.frame10.frame4 -height {30} -width {30}
```

```
# build widget .query.frame10.frame4.label0
label .query.frame10.frame4.label0 -borderwidth {0} -font {-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1} -text {}
```

```
# build widget .query.frame10.frame4.canvas1
canvas .query.frame10.frame4.canvas1 -borderwidth {1} -height {21} -highlightthickness {0} -relief {raised} -width {12}
```

```
# bindings
```

```
bind .query.frame10.frame4.canvas1 <Button-1> {%W configure -relief sunken
incr_index qindex(1)}
bind .query.frame10.frame4.canvas1 <ButtonRelease-1> {%W configure -relief raised}
```

```
# build widget .query.frame10.frame4.canvas2
canvas .query.frame10.frame4.canvas2 -borderwidth {1} -height {21} -highlightthickness {0} -relief {raised} -width {12}
```

```
# bindings
```

```
bind .query.frame10.frame4.canvas2 <Button-1> {%W configure -relief sunken
incr_index qindex(2)}
bind .query.frame10.frame4.canvas2 <ButtonRelease-1> {%W configure -relief raised}
```

```

# build widget .query.frame10.frame4.canvas3
canvas .query.frame10.frame4.canvas3 -borderwidth {1} -height {21} -highlightthickness {0} -relief {raised} -width {12}
# bindings
bind .query.frame10.frame4.canvas3 <Button-1> {%W configure -relief sunken
incr_index qindex(3)}
bind .query.frame10.frame4.canvas3 <ButtonRelease-1> {%W configure -relief raised}

# build widget .query.frame10.frame4.canvas4
canvas .query.frame10.frame4.canvas4 -borderwidth {1} -height {21} -highlightthickness {0} -relief {raised} -width {12}
# bindings
bind .query.frame10.frame4.canvas4 <Button-1> {%W configure -relief sunken
decr_index qindex(1)}
bind .query.frame10.frame4.canvas4 <ButtonRelease-1> {%W configure -relief raised}

# build widget .query.frame10.frame4.canvas5
canvas .query.frame10.frame4.canvas5 -borderwidth {1} -height {21} -highlightthickness {0} -relief {raised} -width {12}
# bindings
bind .query.frame10.frame4.canvas5 <Button-1> {%W configure -relief sunken
decr_index qindex(2)}
bind .query.frame10.frame4.canvas5 <ButtonRelease-1> {%W configure -relief raised}

# build widget .query.frame10.frame4.canvas6
canvas .query.frame10.frame4.canvas6 -borderwidth {1} -height {21} -highlightthickness {0} -relief {raised} -width {12}
# bindings
bind .query.frame10.frame4.canvas6 <Button-1> {%W configure -relief sunken
decr_index qindex(3)}
bind .query.frame10.frame4.canvas6 <ButtonRelease-1> {%W configure -relief raised}

# build widget .query.frame10.frame4.entry7
entry .query.frame10.frame4.entry7 -background {lightblue} -textvariable {qindex(1)} -width {5}

# build widget .query.frame10.frame4.entry8
entry .query.frame10.frame4.entry8 -background {lightblue} -textvariable {qindex(2)} -width {5}

# build widget .query.frame10.frame4.entry9
entry .query.frame10.frame4.entry9 -background {lightblue} -textvariable {qindex(3)} -width {5}

# build widget .query.frame10.frame4.label10
label .query.frame10.frame4.label10 -borderwidth {0} -text {, }

# build widget .query.frame10.frame4.label11
label .query.frame10.frame4.label11 -borderwidth {0} -text {, }

# build widget .query.frame10.frame4.label12
label .query.frame10.frame4.label12 -borderwidth {0} -font {-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1} -text {}

# build widget .query.frame10.frame4.label5
label .query.frame10.frame4.label5 -borderwidth {0} -font {-adobe-helvetica-bold-o-normal--12-120-75-75-p-69-iso8859-1} -text {Array index: }

# pack master .query.frame
pack configure .query.frame.scrollbar2 -fill y -side right
pack configure .query.frame.listbox1 -expand 1 -fill both
pack configure .query.frame.scrollbar3 -fill x -side bottom

# pack master .query.frame3
pack configure .query.frame3.button4 -side left

# pack master .query.frame10
pack configure .query.frame10.frame2 -fill x
pack configure .query.frame10.frame4 -fill x

# pack master .query.frame10.frame2
pack configure .query.frame10.frame2.label5 -side left
pack configure .query.frame10.frame2.radiobutton12 -expand 1 -fill x -side left
pack configure .query.frame10.frame2.radiobutton13 -expand 1 -fill x -side left
pack configure .query.frame10.frame2.entry6 -side left

# pack master .query.frame10.frame4
pack configure .query.frame10.frame4.label5 -side left
pack configure .query.frame10.frame4.label0 -side left
pack configure .query.frame10.frame4.canvas4 -side left
pack configure .query.frame10.frame4.entry7 -side left
pack configure .query.frame10.frame4.canvas1 -side left
pack configure .query.frame10.frame4.label10 -side left
pack configure .query.frame10.frame4.canvas5 -side left

```



```

pack configure .query.frame10.frame4.entry8 -side left
pack configure .query.frame10.frame4.canvas2 -side left
pack configure .query.frame10.frame4.label11 -side left
pack configure .query.frame10.frame4.canvas6 -side left
pack configure .query.frame10.frame4.entry9 -side left
pack configure .query.frame10.frame4.canvas3 -side left
pack configure .query.frame10.frame4.label12 -side left

```

```

# pack master .query
pack configure .query.frame10 -fill x
pack configure .query.frame -expand 1 -fill both
pack configure .query.frame3 -anchor e

```

```

.query.frame.listBox1 insert end {i_1byte      }
.query.frame.listBox1 insert end {i_2byte      }
.query.frame.listBox1 insert end {i_4byte      }
.query.frame.listBox1 insert end {i_4byte_2    temporary 4 byte integer}
.query.frame.listBox1 insert end {i_4byte_3    temporary 4 byte integer}
.query.frame.listBox1 insert end {i_array_2    (2,3)    Two dimension integer array}
.query.frame.listBox1 insert end {i_logical    }
.query.frame.listBox1 insert end {icname      (5)    Name of currently loaded IC}
.query.frame.listBox1 insert end {icrw       IC Read/Write command}

```

```

# build canvas items .query.frame10.frame4.canvas1
set xTmpTag [.query.frame10.frame4.canvas1 create line 7.0 14.0 7.0 6.0]
.query.frame10.frame4.canvas1 itemconfigure $xTmpTag -arrow {last} -joinstyle {miter} -width {3}
# build canvas items .query.frame10.frame4.canvas2
set xTmpTag [.query.frame10.frame4.canvas2 create line 7.0 14.0 7.0 6.0]
.query.frame10.frame4.canvas2 itemconfigure $xTmpTag -arrow {last} -joinstyle {miter} -width {3}
# build canvas items .query.frame10.frame4.canvas3
set xTmpTag [.query.frame10.frame4.canvas3 create line 7.0 14.0 7.0 6.0]
.query.frame10.frame4.canvas3 itemconfigure $xTmpTag -arrow {last} -joinstyle {miter} -width {3}
# build canvas items .query.frame10.frame4.canvas4
set xTmpTag [.query.frame10.frame4.canvas4 create line 7.0 16.0 7.0 8.0]
.query.frame10.frame4.canvas4 itemconfigure $xTmpTag -arrow {first} -joinstyle {miter} -width {3}
# build canvas items .query.frame10.frame4.canvas5
set xTmpTag [.query.frame10.frame4.canvas5 create line 7.0 16.0 7.0 8.0]
.query.frame10.frame4.canvas5 itemconfigure $xTmpTag -arrow {first} -joinstyle {miter} -width {3}
# build canvas items .query.frame10.frame4.canvas6
set xTmpTag [.query.frame10.frame4.canvas6 create line 7.0 16.0 7.0 8.0]
.query.frame10.frame4.canvas6 itemconfigure $xTmpTag -arrow {first} -joinstyle {miter} -width {3}

```

```

if {[info procs XFEdit] != ""} {
    catch "XFMiscBindWidgetTree .query"
    after 2 *catch {XFEditSetShowWindows}
}
}

```

```

proc DestroyWindow.query {} {# xf ignore me 7
if {[info procs XFEdit] != ""} {
    if {[info commands .query] != ""} {
        global xfShowWindow.query
        set xfShowWindow.query 0
        XFEditSetPath .
        after 2 *XFSaveAsProc .query; XFEditSetShowWindows"
    }
} {
    catch "destroy .query"
    update
}
}

```

```

# procedure to show window ShowWindow.rfi
proc ShowWindow.rfi { args } {
# xf ignore me 7

```

```

# build widget .rfi
if {[info procs XFEdit] != ""} {
    catch "XFDestroy .rfi"
} {
    catch "destroy .rfi"
}
}
tooplevel .rfi -relief {raised}

```

```

# Window manager configurations
wm positionfrom .rfi program
wm sizefrom .rfi program
wm maxsize .rfi 1000 864
wm minsize .rfi 510 300
wm protocol .rfi WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .rfi {Remote Function Index}

# bindings
bind .rfi <Reparent> {rfi_fill " "}

# build widget .rfi.frame
frame .rfi.frame

# build widget .rfi.frame.scrollbar2
scrollbar .rfi.frame.scrollbar2 -borderwidth {1} -command {rfi.frame.listbox1 yview} -highlightthickness {0} -relief {raised}

# build widget .rfi.frame.listbox1
listbox .rfi.frame.listbox1 -background {#c8c8c8} -borderwidth {1} -font {-adobe-courier-bold-r-normal-12-120-75-75-m-70-iso8859-1} -yscrollcommand
{rfi.frame.scrollbar2 set}
# bindings
bind .rfi.frame.listbox1 <Double-Button-1> {set index [%W nearest %y]
if {$index != ""} {
    rfp [lindex [%W get $index] 0]
}}

# build widget .rfi.frame.frame30
frame .rfi.frame.frame30 -height {30} -width {30}

# build widget .rfi.frame.frame30.label31
label .rfi.frame.frame30.label31 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal-12-120-75-75-p-70-iso8859-1} -foreground {white} -relief
{raised} -text {RF Name} -width {20}

# build widget .rfi.frame.frame30.label32
label .rfi.frame.frame30.label32 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal-12-120-75-75-p-70-iso8859-1} -foreground {white} -relief
{raised} -text {Range} -width {18}

# build widget .rfi.frame.frame30.label33
label .rfi.frame.frame30.label33 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal-12-120-75-75-p-70-iso8859-1} -foreground {white} -relief
{raised} -text {Description} -width {15}

# build widget .rfi.frame.frame30.label34
label .rfi.frame.frame30.label34 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-bold-r-normal-12-120-75-75-p-70-iso8859-1} -foreground {white} -relief
{raised} -text {Status} -width {12}

# build widget .rfi.frame.frame35
frame .rfi.frame.frame35 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .rfi.frame.frame35.label38
label .rfi.frame.frame35.label38 -background {slategrey} -borderwidth {1} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {white} -relief
{raised} -text {Search: }

# build widget .rfi.frame.frame35.entry39
entry .rfi.frame.frame35.entry39 -background {lightblue} -borderwidth {1} -font {-adobe-helvetica-medium-o-normal-14-140-75-75-p-78-iso8859-1} -highlightthickness {0} -width
{10}
# bindings
bind .rfi.frame.frame35.entry39 <Key-Return> {rfi_fill [%W get]}

# build widget .rfi.frame.frame24
frame .rfi.frame.frame24 -height {30} -width {30}

# build widget .rfi.frame.frame24.button36
button .rfi.frame.frame24.button36 -borderwidth {1} -command {set index [.rfi.frame.listbox1 curselection]
if {$index != ""} {
    rfp [lindex [.rfi.frame.listbox1 get $index] 0]
}} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -text {Edit}

# build widget .rfi.frame.frame24.button37
button .rfi.frame.frame24.button37 -borderwidth {1} -command {rfi.frame.listbox1 delete 0 end
DestroyWindow .rfi} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -padx {11} -pady {4} -state {active} -text {Done}

# pack master .rfi.frame
pack configure .rfi.frame.scrollbar2 -fill y -side right

```

```

pack configure .rfi.frame.frame30 -fill x
pack configure .rfi.frame.listbox1 -expand 1 -fill both

# pack master .rfi.frame.frame30
pack configure .rfi.frame.frame30.label31 -side left
pack configure .rfi.frame.frame30.label34 -side left
pack configure .rfi.frame.frame30.label32 -side left
pack configure .rfi.frame.frame30.label33 -expand 1 -fill x -side left

# pack master .rfi.frame35
pack configure .rfi.frame35.label38 -fill y -side left
pack configure .rfi.frame35.entry39 -expand 1 -fill both -side left

# pack master .rfi.frame24
pack configure .rfi.frame24.button37 -expand 1 -fill x -side left
pack configure .rfi.frame24.button36 -expand 1 -fill x -side left

# pack master .rfi
pack configure .rfi.frame -expand 1 -fill both
pack configure .rfi.frame35 -fill x
pack configure .rfi.frame24 -anchor e -side bottom

.rfi.frame35.entry39 insert end {}

```

```

if {[info procs XFEdit] != ""} {
  catch "XFMiscBindWidgetTree .rfi"
  after 2 "catch {XFEditSetShowWindows}"
}
}

```

```

proc DestroyWindow.rfi {} {# xf ignore me 7
  if {[info procs XFEdit] != ""} {
    if {[info commands .rfi] != ""} {
      global xfShowWindow.rfi
      set xfShowWindow.rfi 0
      XFEditSetPath .
      after 2 "XFSaveAsProc .rfi; XFEditSetShowWindows"
    }
  }
  catch "destroy .rfi"
  update
}
}

```

```

# procedure to show window ShowWindow.rfp
proc ShowWindow.rfp { args } {
  # xf ignore me 7

```

```

# build widget .rfp
if {[info procs XFEdit] != ""} {
  catch "XFDestroy .rfp"
}
catch "destroy .rfp"
}
}
toplevel .rfp -relief {raised}

```

```

# Window manager configurations
wm positionfrom .rfp program
wm sizefrom .rfp program
wm maxsize .rfp 1024 768
wm minsize .rfp 240 138
wm protocol .rfp WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .rfp {Remote Function Control}

```

```

# build widget .rfp.label1
label .rfp.label1 -background {grey60} -font {-adobe-helvetica-medium-r-normal-18-180-75-75-p-98-iso8859-1} -foreground {yellow} -padx {8} -pady {4} -relief {ridge}
-text {Heater 2 control switch} -textvariable {rfpdesc}

```

```

# build widget .rfp.frame2
frame .rfp.frame2 -background {grey50} -height {30} -width {30}

```

```

# build widget .rfp.frame2.frame3
frame .rfp.frame2.frame3 -background {grey50} -height {30} -width {30}

# build widget .rfp.frame2.frame3.frame4
frame .rfp.frame2.frame3.frame4 -background {grey50} -height {30} -width {30}

# build widget .rfp.frame2.frame3.frame4.label6
label .rfp.frame2.frame3.frame4.label6 -background {grey50} -borderwidth {0} -foreground {white} -text {Actual } -width {8}

# build widget .rfp.frame2.frame3.frame4.label7
label .rfp.frame2.frame3.frame4.label7 -background {grey50} -borderwidth {0} -foreground {white} -text {Requested}

# build widget .rfp.frame2.frame3.frame5
frame .rfp.frame2.frame3.frame5 -background {grey50} -height {30} -width {30}

# build widget .rfp.frame2.frame3.frame5.label9
label .rfp.frame2.frame3.frame5.label9 -background {grey50} -borderwidth {0} -text {off} -textvariable {rfpfeedback} -width {7}

# build widget .rfp.frame2.frame3.frame5.entry10
entry .rfp.frame2.frame3.frame5.entry10 -background {azure} -highlightthickness {0} -state {disabled} -textvariable {rfp_requested} -width {10}
# bindings
bind .rfp.frame2.frame3.frame5.entry10 <Key-Return> {rfp_set $rfp_requested}

# build widget .rfp.frame2.frame3.button11
button .rfp.frame2.frame3.button11 -borderwidth {1} -command {set index [.rfp.frame2.frame.listbox1 curselection]}
if {$index != ""} {
    rfp_set [.rfp.frame2.frame.listbox1 get $index]
} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -highlightbackground {grey50} -padx {11} -pady {4} -text {Apply} -width {4}

# build widget .rfp.frame2.frame3.button13
button .rfp.frame2.frame3.button13 -borderwidth {1} -command {DestroyWindow.rfp} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1}
-highlightbackground {grey50} -padx {11} -pady {4} -state {active} -text {Done} -width {4}

# build widget .rfp.frame2.frame3.frame3
frame .rfp.frame2.frame3.frame3 -height {30} -width {30}

# build widget .rfp.frame2.frame3.frame3.label1
label .rfp.frame2.frame3.frame3.label1 -background {grey50} -borderwidth {0} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -textvariable {rfpjoin}
-width {2}

# build widget .rfp.frame2.frame3.frame3.label2
label .rfp.frame2.frame3.frame3.label2 -background {grey50} -borderwidth {0} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -foreground {green}
-textvariable {rfpmin} -width {6}

# build widget .rfp.frame2.frame3.frame3.label4
label .rfp.frame2.frame3.frame3.label4 -background {grey50} -borderwidth {0} -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} -foreground {green}
-textvariable {rfpmax} -width {6}

# build widget .rfp.frame2.frame
frame .rfp.frame2.frame

# build widget .rfp.frame2.frame.scrollbar2
scrollbar .rfp.frame2.frame.scrollbar2 -borderwidth {1} -command {rfp.frame2.frame.listbox1 yview} -highlightthickness {0} -relief {raised}

# build widget .rfp.frame2.frame.listbox1
listbox .rfp.frame2.frame.listbox1 -background {#c8c8c8} -borderwidth {1} -yscrollcommand {rfp.frame2.frame.scrollbar2 set}

# pack master .rfp.frame2
pack configure .rfp.frame2.frame3 -fill x -side left
pack configure .rfp.frame2.frame -expand 1 -fill both -side left

# pack master .rfp.frame2.frame3
pack configure .rfp.frame2.frame3.frame4 -fill x
pack configure .rfp.frame2.frame3.frame5 -ipady 4
pack configure .rfp.frame2.frame3.frame3
pack configure .rfp.frame2.frame3.button11
pack configure .rfp.frame2.frame3.button13

# pack master .rfp.frame2.frame3.frame4
pack configure .rfp.frame2.frame3.frame4.label6 -side left
pack configure .rfp.frame2.frame3.frame4.label7 -side left

# pack master .rfp.frame2.frame3.frame5
pack configure .rfp.frame2.frame3.frame5.label9 -side left

```

```
pack configure .rfp.frame2.frame3.frame5.entry10 -padx 4 -side right
```

```
# pack master .rfp.frame2.frame3.frame3
pack configure .rfp.frame2.frame3.frame3.label2 -side left
pack configure .rfp.frame2.frame3.frame3.label1 -side left
pack configure .rfp.frame2.frame3.frame3.label4 -side right
```

```
# pack master .rfp.frame2.frame
pack configure .rfp.frame2.frame.scrollbar2 -fill y -side right
pack configure .rfp.frame2.frame.listbox1 -expand 1 -fill both
```

```
# pack master .rfp
pack configure .rfp.label1 -fill x
pack configure .rfp.frame2 -expand 1 -fill both
```

```
.rfp.frame2.frame.listbox1 insert end {off}
.rfp.frame2.frame.listbox1 insert end {on}
```

```
if {[info procs XFEdit] != ""} {
  catch *XFMiscBindWidgetTree .rfp*
  after 2 *catch {XFEditSetShowWindows}*
}
}
```

```
proc DestroyWindow.rfp {} {# xf ignore me 7
if {[info procs XFEdit] != ""} {
  if {[info commands .rfp] != ""} {
    global xfShowWindow.rfp
    set xfShowWindow.rfp 0
    XFEditSetPath .
    after 2 *XFSaveAsProc .rfp; XFEditSetShowWindows*
  }
} {
  catch "destroy .rfp"
  update
}
}
```

```
# procedure to show window .
proc ShowWindow. {args} {# xf ignore me 7
```

```
# Window manager configurations
wm positionfrom . user
wm sizefrom . ""
wm maxsize . 1152 847
wm minsize . 132 1
wm protocol . WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path." to show/hide windows.}}
wm title . {{asus): s:\demo\rtsched.exe}
```

```
# build widget .frame12
frame .frame12 \
  -height {30} \
  -width {30}
```

```
# build widget .frame12.frame0
frame .frame12.frame0 \
  -height {30} \
  -relief {raised} \
  -width {30}
```

```
# build widget .frame12.frame0.ctrbtn1
frame .frame12.frame0.ctrbtn1 \
  -relief {raised}
```

```
# build widget .frame12.frame0.ctrbtn1.menubutton8
menubutton .frame12.frame0.ctrbtn1.menubutton8 \
  -background {#595959} \
  -font {-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1} \
  -foreground {yellow} \
  -menu { .frame12.frame0.ctrbtn1.menubutton8.m} \
```

```

-padx {5} \
-pady {3} \
-relief {raised} \
-text {Status} \
-width {8}

# build widget .frame12.frame0.ctribtn1.menubutton8.m
menu .frame12.frame0.ctribtn1.menubutton8.m \
  -borderwidth {0} \
  -tearoff {0}
.frame12.frame0.ctribtn1.menubutton8.m add command \
  -command {css_cmd recd run} \
  -label {Run}
.frame12.frame0.ctribtn1.menubutton8.m add command \
  -command {if {$sics_confirm == "on"} {
    set ans [tk_messageBox -icon question -type yesno -message "Do you really want to go to FREEZE?"]
  } { set ans yes }

if {$ans == "yes"} {css_cmd recd frz} \
  -label {Freeze}
.frame12.frame0.ctribtn1.menubutton8.m add separator
.frame12.frame0.ctribtn1.menubutton8.m add command \
  -command {if {$sics_confirm == "on"} {
    set ans [tk_messageBox -icon question -type yesno -message "Do you really want to RESET?"]
  } { set ans yes }

if {$ans == "yes"} {css_cmd recd "reset $curIC"} \
  -label {Reset}
.frame12.frame0.ctribtn1.menubutton8.m add separator
.frame12.frame0.ctribtn1.menubutton8.m add command \
  -command {if {$sics_confirm == "on"} {
    set ans [tk_messageBox -icon question -type yesno -message "Do you really want to go to TERMINATE the\ncurrent real time program and RELOAD it?"]
  } { set ans yes }

if {$ans == "yes"} {
  css_cmd recd unload
  css_cmd recd load
  after 2000 {ics_titlebar_update}
} \
-label {Load}
.frame12.frame0.ctribtn1.menubutton8.m add command \
  -command {if {$sics_confirm == "on"} {
    set ans [tk_messageBox -icon question -type yesno -message "Do you really want to go to TERMINATE\nthe current real time program?"]
  } { set ans yes }

if {$ans == "yes"} {css_cmd recd unload} \
  -label {Unload}
.frame12.frame0.ctribtn1.menubutton8.m add separator
.frame12.frame0.ctribtn1.menubutton8.m add command \
  -command {ShowWindow.pref} \
  -label {Preferences}
.frame12.frame0.ctribtn1.menubutton8.m add separator
.frame12.frame0.ctribtn1.menubutton8.m add command \
  -command {ics_quit} \
  -label {Exit}

# build widget .frame12.frame0.ctribtn1.menubutton11
menubutton .frame12.frame0.ctribtn1.menubutton11 \
  -background {#595959} \
  -font {-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1} \
  -foreground {yellow} \
  -menu {frame12.frame0.ctribtn1.menubutton11.m} \
  -padx {5} \
  -pady {3} \
  -relief {raised} \
  -text {Rate: 1.00} \
  -textvariable {rates1f} \
  -width {8}

# build widget .frame12.frame0.ctribtn1.menubutton11.m
menu .frame12.frame0.ctribtn1.menubutton11.m \
  -borderwidth {0} \
  -tearoff {0}
.frame12.frame0.ctribtn1.menubutton11.m add cascade \
  -label {Slow} \

```

```
-menu {frame12.frame0.ctribn1.menubutton11.m.m}
.frame12.frame0.ctribn1.menubutton11.m add command \
-command {css_cmd recd "slow 1"} \
-label {Real time}
.frame12.frame0.ctribn1.menubutton11.m add cascade \
-label {Fast} \
-menu {frame12.frame0.ctribn1.menubutton11.m.m2}
```

```
# build widget .frame12.frame0.ctribn1.menubutton11.m.m
menu .frame12.frame0.ctribn1.menubutton11.m.m \
-borderwidth {0} \
-tearoff {0}
.frame12.frame0.ctribn1.menubutton11.m.m add command \
-command {css_cmd recd "slow 2"} \
-label {1/2}
.frame12.frame0.ctribn1.menubutton11.m.m add command \
-command {css_cmd recd "slow 3"} \
-label {1/3}
.frame12.frame0.ctribn1.menubutton11.m.m add command \
-command {css_cmd recd "slow 4"} \
-label {1/4}
.frame12.frame0.ctribn1.menubutton11.m.m add command \
-command {css_cmd recd "slow 5"} \
-label {1/5}
.frame12.frame0.ctribn1.menubutton11.m.m add command \
-command {css_cmd recd "slow 6"} \
-label {1/6}
.frame12.frame0.ctribn1.menubutton11.m.m add command \
-command {css_cmd recd "slow 7"} \
-label {1/7}
.frame12.frame0.ctribn1.menubutton11.m.m add command \
-command {css_cmd recd "slow 8"} \
-label {1/8}
.frame12.frame0.ctribn1.menubutton11.m.m add command \
-command {css_cmd recd "slow 9"} \
-label {1/9}
.frame12.frame0.ctribn1.menubutton11.m.m add command \
-command {css_cmd recd "slow 10"} \
-label {1/10}
```

```
# build widget .frame12.frame0.ctribn1.menubutton11.m.m2
menu .frame12.frame0.ctribn1.menubutton11.m.m2 \
-borderwidth {0} \
-tearoff {0}
.frame12.frame0.ctribn1.menubutton11.m.m2 add command \
-command {css_cmd recd "fast 2"} \
-label {2}
.frame12.frame0.ctribn1.menubutton11.m.m2 add command \
-command {css_cmd recd "fast 3"} \
-label {3}
.frame12.frame0.ctribn1.menubutton11.m.m2 add command \
-command {css_cmd recd "fast 4"} \
-label {4}
.frame12.frame0.ctribn1.menubutton11.m.m2 add command \
-command {css_cmd recd "fast 5"} \
-label {5}
.frame12.frame0.ctribn1.menubutton11.m.m2 add command \
-command {css_cmd recd "fast 6"} \
-label {6}
.frame12.frame0.ctribn1.menubutton11.m.m2 add command \
-command {css_cmd recd "fast 7"} \
-label {7}
.frame12.frame0.ctribn1.menubutton11.m.m2 add command \
-command {css_cmd recd "fast 8"} \
-label {8}
.frame12.frame0.ctribn1.menubutton11.m.m2 add command \
-command {css_cmd recd "fast 9"} \
-label {9}
.frame12.frame0.ctribn1.menubutton11.m.m2 add command \
-command {css_cmd recd "fast 10"} \
-label {10}
```

```
# build widget .frame12.frame0.ctribn2
frame .frame12.frame0.ctribn2 \
-relief {raised}
```

```

# build widget .frame12.frame0.ctrbtn2.menubutton9
menubutton .frame12.frame0.ctrbtn2.menubutton9 \
-background {#595959} \
-font {-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1} \
-foreground {yellow} \
-menu { .frame12.frame0.ctrbtn2.menubutton9.m } \
-padx {5} \
-pady {3} \
-relief {raised} \
-text {Diagrams} \
-width {8}

```

```

# build widget .frame12.frame0.ctrbtn2.menubutton9.m
menu .frame12.frame0.ctrbtn2.menubutton9.m \
-borderwidth {0} \
-tearoff {0}
.frame12.frame0.ctrbtn2.menubutton9.m add command \
-command {ShowWindow.diagidx} \
-label {Diagram Index}
.frame12.frame0.ctrbtn2.menubutton9.m add cascade \
-command {
} \
-label {Electrical Distribution} \
-menu { .frame12.frame0.ctrbtn2.menubutton9.m.m }

```

```

# build widget .frame12.frame0.ctrbtn2.menubutton9.m.m
menu .frame12.frame0.ctrbtn2.menubutton9.m.m \
-borderwidth {0} \
-tearoff {0}
.frame12.frame0.ctrbtn2.menubutton9.m.m add command \
-command {showdiag ed1.tcl}
} \
-label {ED1}
.frame12.frame0.ctrbtn2.menubutton9.m.m add command \
-command {showdiag ed2.tcl}
} \
-label {ED2}

```

```

# build widget .frame12.frame0.ctrbtn2.menubutton10
menubutton .frame12.frame0.ctrbtn2.menubutton10 \
-background {#595959} \
-font {-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1} \
-foreground {yellow} \
-menu { .frame12.frame0.ctrbtn2.menubutton10.m } \
-padx {5} \
-pady {3} \
-relief {raised} \
-text {Tableaus} \
-width {8}

```

```

# build widget .frame12.frame0.ctrbtn2.menubutton10.m
menu .frame12.frame0.ctrbtn2.menubutton10.m \
-borderwidth {0} \
-tearoff {0}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {ShowWindow.alarm} \
-label {Alarm Monitor}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {ShowWindow.aps} \
-label {Automated Procedures}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {ShowWindow.appmsg} \
-label {Automated Procedure Messages}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {ShowWindow.backtrack} \
-label {Backtrack / Replay}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {ShowWindow.evs} \
-label {Event Summary}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {ShowWindow.ets} \
-label {Event Triggers}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {ShowWindow.mfi} \

```



```

-label {Malfunction Index}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {ShowWindow.mfs} \
-label {Malfunction Status}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {ShowWindow.monparm}
} \
-label {Monitored Parameters}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {exec wish icsgraph.tcl &} \
-label {Plotting}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {ShowWindow.rfi} \
-label {Remote Function Index}
.frame12.frame0.ctrbtn2.menubutton10.m add command \
-command {vcs_show} \
-label {Virtual Control Station}

# build widget .frame12.frame15
frame .frame12.frame15 \
-height {30} \
-width {30}

# build widget .frame12.frame15.frame20
frame .frame12.frame15.frame20 \
-height {30} \
-width {30}

# build widget .frame12.frame15.frame20.frame22
frame .frame12.frame15.frame20.frame22 \
-height {30} \
-width {30}

# build widget .frame12.frame15.frame20.frame22.label26
label .frame12.frame15.frame20.frame22.label26 \
-background {grey75} \
-relief {groove} \
-text {12/09/97} \
-textvariable {datestr} \
-width {8}

# build widget .frame12.frame15.frame20.frame23
frame .frame12.frame15.frame20.frame23 \
-height {30} \
-width {30}

# build widget .frame12.frame15.frame20.frame23.label27
label .frame12.frame15.frame20.frame23.label27 \
-background {grey75} \
-relief {groove} \
-text {19:49:37} \
-textvariable {clockstr} \
-width {8}

# build widget .frame12.frame15.frame21
frame .frame12.frame15.frame21 \
-height {30} \
-width {30}

# build widget .frame12.frame15.frame21.frame24
frame .frame12.frame15.frame21.frame24 \
-height {30} \
-width {30}

# build widget .frame12.frame15.frame21.frame24.label28
label .frame12.frame15.frame21.frame24.label28 \
-background {grey75} \
-relief {groove} \
-text {0.000000} \
-textvariable {cpusage} \
-width {8}

# build widget .frame12.frame15.frame21.frame25
frame .frame12.frame15.frame21.frame25 \
-height {30} \

```

```

-width {30}

# build widget .frame12.frame15.frame21.frame25.label29
label .frame12.frame15.frame21.frame25.label29 \
-background {grey75} \
-relief {groove} \
-text {01:37:33} \
-textvariable {simtime} \
-width {8}

# build widget .frame12.frame15.frame30
frame .frame12.frame15.frame30 \
-height {30} \
-width {30}

# build widget .frame12.frame15.frame30.label32
label .frame12.frame15.frame30.label32 \
-background {grey75} \
-relief {groove} \
-text {mal: 0:0} \
-textvariable {ics_malfsta} \
-width {8}

# build widget .frame12.frame15.frame31
frame .frame12.frame15.frame31 \
-height {30} \
-width {30}

# build widget .frame12.frame15.frame31.label33
label .frame12.frame15.frame31.label33 \
-background {grey75} \
-relief {groove} \
-width {8}

# build widget .frame12.frame16
frame .frame12.frame16 \
-background {systembuttonface} \
-borderwidth {1} \
-relief {raised}
# bindings
bind .frame12.frame16 <Enter> {%W configure -bg systembuttonhighlight}
bind .frame12.frame16 <Leave> {%W configure -bg systembuttonface}

# build widget .frame12.frame16.label17
label .frame12.frame16.label17 \
-background {systembuttonface} \
-borderwidth {0} \
-font {-adobe-helvetica-medium-r-normal--18-180-75-75-p-98-iso8859-1} \
-foreground {blue} \
-text {{C}}
# bindings
bind .frame12.frame16.label17 <Button-1> {ShowWindow.icsum}
bind .frame12.frame16.label17 <Enter> {.frame12.frame16.label17 configure -bg systembuttonhighlight}
.frame12.frame16.label18 configure -bg systembuttonhighlight
.frame12.frame16.label19 configure -bg systembuttonhighlight}
bind .frame12.frame16.label17 <Leave> {.frame12.frame16.label17 configure -bg systembuttonface}
.frame12.frame16.label18 configure -bg systembuttonface}
.frame12.frame16.label19 configure -bg systembuttonface}

# build widget .frame12.frame16.label18
label .frame12.frame16.label18 \
-background {systembuttonface} \
-borderwidth {0} \
-foreground {red} \
-text {{C50}} \
-textvariable {curC}
# bindings
bind .frame12.frame16.label18 <Button-1> {ShowWindow.icsum}
bind .frame12.frame16.label18 <Enter> {.frame12.frame16.label17 configure -bg systembuttonhighlight}
.frame12.frame16.label18 configure -bg systembuttonhighlight}
.frame12.frame16.label19 configure -bg systembuttonhighlight}
bind .frame12.frame16.label18 <Leave> {.frame12.frame16.label17 configure -bg systembuttonface}
.frame12.frame16.label18 configure -bg systembuttonface}
.frame12.frame16.label19 configure -bg systembuttonface}

```

```

# build widget .frame12.frame16.label19
label .frame12.frame16.label19 \
  -background {systembuttonface} \
  -borderwidth {0} \
  -padx {0} \
  -pady {0} \
  -text {Run} \
  -textvariable {status} \
  -width {7}
# bindings
bind .frame12.frame16.label19 <Button-1> {ShowWindow.icsum}
bind .frame12.frame16.label19 <Enter> {frame12.frame16.label17 configure -bg systembuttonhighlight
.frame12.frame16.label18 configure -bg systembuttonhighlight
.frame12.frame16.label19 configure -bg systembuttonhighlight}
bind .frame12.frame16.label19 <Leave> {frame12.frame16.label17 configure -bg systembuttonface
.frame12.frame16.label18 configure -bg systembuttonface
.frame12.frame16.label19 configure -bg systembuttonface}

```

```

# pack master .frame12
pack configure .frame12.frame15 \
  -fill y \
  -side left
pack configure .frame12.frame16 \
  -fill y \
  -ipady 3 \
  -side left
pack configure .frame12.frame0 \
  -fill y \
  -side left

```

```

# pack master .frame12.frame0
pack configure .frame12.frame0.cribtn1 \
  -expand 1 \
  -fill y
pack configure .frame12.frame0.cribtn2 \
  -expand 1 \
  -fill y \
  -side bottom

```

```

# pack master .frame12.frame0.cribtn1
pack configure .frame12.frame0.cribtn1.menubutton8 \
  -expand 1 \
  -fill y \
  -side left
pack configure .frame12.frame0.cribtn1.menubutton11 \
  -expand 1 \
  -fill y

```

```

# pack master .frame12.frame0.cribtn2
pack configure .frame12.frame0.cribtn2.menubutton9 \
  -fill y \
  -side left
pack configure .frame12.frame0.cribtn2.menubutton10 \
  -fill y \
  -side left

```

```

# pack master .frame12.frame15
pack configure .frame12.frame15.frame20
pack configure .frame12.frame15.frame21
pack configure .frame12.frame15.frame30 \
  -side left
pack configure .frame12.frame15.frame31 \
  -side left

```

```

# pack master .frame12.frame15.frame20
pack configure .frame12.frame15.frame20.frame22 \
  -side left
pack configure .frame12.frame15.frame20.frame23 \
  -side left

```

```

# pack master .frame12.frame15.frame20.frame22
pack configure .frame12.frame15.frame20.frame22.label26

```

```

# pack master .frame12.frame15.frame20.frame23
pack configure .frame12.frame15.frame20.frame23.label27

```

```

# pack master .frame12.frame15.frame21
pack configure .frame12.frame15.frame21.frame24 \
-side left
pack configure .frame12.frame15.frame21.frame25 \
-side left

# pack master .frame12.frame15.frame21.frame24
pack configure .frame12.frame15.frame21.frame24.label28

# pack master .frame12.frame15.frame21.frame25
pack configure .frame12.frame15.frame21.frame25.label29

# pack master .frame12.frame15.frame30
pack configure .frame12.frame15.frame30.label32

# pack master .frame12.frame15.frame31
pack configure .frame12.frame15.frame31.label33

# pack master .frame12.frame16
pack configure .frame12.frame16.label17 \
-fill x
pack configure .frame12.frame16.label18 \
-fill x
pack configure .frame12.frame16.label19 \
-expand 1 \
-fill both

# pack master .
pack configure .frame12

if {[info procs XFEdit] != ""} {
  catch "XFMiscBindWidgetTree ."
  after 2 "catch {XFEditSetShowWindows}"
}
}

# User defined procedures

# Procedure: backtrack_getstatus
proc backtrack_getstatus {} {
  if {[isdisplayed .backtrack]} {
    set blist [css_cmd stat "bt_info"]
    set orgsel [.backtrack.frame.listbox1 curselection]
    set orgidx [.backtrack.frame.listbox1 nearest 0]
    .backtrack.frame.listbox1 delete 0 end
    eval ".backtrack.frame.listbox1 insert end $blist"
    .backtrack.frame.listbox1 yview $orgidx
    if {$orgsel != ""} {.backtrack.frame.listbox1 selection set $orgsel}
  }
}

# Procedure: css_stat
proc css_stat { cmd } {
  css_cmd stat $cmd
}

# Procedure: cyclic_user
proc cyclic_user {} {
  global clockstr
  global datestr
  global status
  global simtime
  global simsecs
  global cpusage
  global curIC
  global loadname
  global ratestr
  global stathost
  global bt_interval
  global ics_malfstat

```

```

getval_cyclic
set clockstr [clock format [clock seconds] -format %X]
set datestr [clock format [clock seconds] -format %x]
set info [css_cmd stat status]
set status [lindex $info 0]
set rate [lindex $info 1]
set ratesr "Rate: $rate"
set simlme [lindex $info 2]
set simsecs [lindex $info 3]
set cpusage [lindex $info 4]
set loadname [file tail [lindex $info 5]]
set curlC [lindex $info 6]
set bl_interval [lindex $info 7]
set ics_malfstat [lindex $info 8]
mps_cyclic
disp_getstatus [info host]
backtrack_getstatus
mfs_getstatus
vcs_getstatus
}

```

```

# Procedure: decr_index
proc decr_index {iname} {
  upvar $iname index
  if {[catch {incr index -1}] } {set index 1}
  if {$index < 1} {set index 1}
}

```

```

# Procedure: disp_getstatus
proc disp_getstatus {host} {
  global stathost

```

```

  if {[winfo exists .dispstat.frame.listBox1]} {
    set stathost "Connections from $host"
    set slist [css_cmd stat "getdisp $host"]
    .dispstat.frame.listBox1 delete 0 end
    foreach item $slist {
      set i1 [split [lindex $item 0] :]
      set ihost [lindex $i1 0]
      if {$ihost == $host} {
        set a1 [lindex $i1 1]
        set a2 [lindex $i1 2]
        set a3 [lindex $i1 3]
        set line [format "%-16s $a2 $a3" $a1]
        .dispstat.frame.listBox1 insert end $line
      }
    }
  }
}

```

```

# Procedure: drop_dragwin
proc drop_dragwin {} {
  global dragbuffer

  set dragbuffer [.dragwin.buffer cget -text]
  if {"[info procs XFEdit]" != ""} {
    if {"[info commands .dragwin]" != ""} {
      XFDestroy .dragwin
    }
  }
  catch "destroy .dragwin"
  update
  after 100 {set dragbuffer ""; set dragsrc ""}
}

```

```

# Procedure: history
proc history {args} {
  set len [length $args]
  if {$len == 0} {

```

```

    return [tcl::HistInfo]
}
set key [lindex $args 0]
set options "add, change, clear, event, info, keep, nextid, or redo"
switch -glob - $key {
    a* { # history add

        if {$len > 3} {
            return -code error "wrong # args: should be \"history add event ?exec?\""
        }
        if {[string match $key* add]} {
            return -code error "bad option \"$key\": must be $options"
        }
        if {$len == 3} {
            set arg [lindex $args 2]
            if {!(string match e* $arg) && (string match $arg* exec)} {
                return -code error "bad argument \"$arg\": should be \"exec\""
            }
        }
        return [tcl::HistAdd [lindex $args 1] [lindex $args 2]]
    }
    ch* { # history change

        if {($len > 3) || ($len < 2)} {
            return -code error "wrong # args: should be \"history change newValue ?event?\""
        }
        if {[string match $key* change]} {
            return -code error "bad option \"$key\": must be $options"
        }
        if {$len == 2} {
            set event 0
        } else {
            set event [lindex $args 2]
        }

        return [tcl::HistChange [lindex $args 1] $event]
    }
    cl* { # history clear

        if {$len > 1} {
            return -code error "wrong # args: should be \"history clear\""
        }
        if {[string match $key* clear]} {
            return -code error "bad option \"$key\": must be $options"
        }
        return [tcl::HistClear]
    }
    e* { # history event

        if {$len > 2} {
            return -code error "wrong # args: should be \"history event ?event?\""
        }
        if {[string match $key* event]} {
            return -code error "bad option \"$key\": must be $options"
        }
        if {$len == 1} {
            set event -1
        } else {
            set event [lindex $args 1]
        }
        return [tcl::HistEvent $event]
    }
    i* { # history info

        if {$len > 2} {
            return -code error "wrong # args: should be \"history info ?count?\""
        }
        if {[string match $key* info]} {
            return -code error "bad option \"$key\": must be $options"
        }
        return [tcl::HistInfo [lindex $args 1]]
    }
    k* { # history keep

        if {$len > 2} {

```

```

        return -code error "wrong # args: should be \"history keep ?count?\""
    }
    if {$len == 1} {
        return [tcl::HistKeep]
    } else {
        set limit [lindex $args 1]
        if {[catch {expr $limit} {}] || ($limit < 0)} {
            return -code error "illegal keep count \"$limit\""
        }
        return [tcl::HistKeep $limit]
    }
}
n* { # history nextid

    if {$len > 1} {
        return -code error "wrong # args: should be \"history nextid\""
    }
    if {[string match $key* nextid]} {
        return -code error "bad option \"$key\": must be $options"
    }
    return [expr $tcl::history(nextid) + 1]
}
r* { # history redo

    if {$len > 2} {
        return -code error "wrong # args: should be \"history redo ?event?\""
    }
    if {[string match $key* redo]} {
        return -code error "bad option \"$key\": must be $options"
    }
    return [tcl::HistRedo [lindex $args 1]]
}
default {
    return -code error "bad option \"$key\": must be $options"
}
}
}

```

Procedure: ics_titlebar_update

```

proc ics_titlebar_update {} {
    global port
    global machine
    global argv0

```

```

# display connection information in window title
set loadpath [lindex [css_cmd stat status] 5]
wm title . "($machine): $loadpath"
}

```

Procedure: icsum_desc

```

proc icsum_desc {} {
    global icselected
    global icnewdesc

```

```

# Update IC description
set index $icselected
incr index -1
set line [.icsum.frame.listBox1 get $index]
set icno [lindex $line 0]
set date [lindex $line 1]
set time [lindex $line 2]
set name [lindex $line 3]
if {[string range $icnewdesc 0 0] == ""} {
    set dstat ""
} {
    set dstat " "
}
set lname [string range $name 0 11]
set item [format " %s %s %s %-12s %s%" $icno $date $time $lname $dstat $icnewdesc]
.icsum.frame.listBox1 delete $index
.icsum.frame.listBox1 insert $index $item

```

Now save to file

```

set fout [open ic/icsum.txt w]
if {$fout != -1} {

```

```

        set size [.icsum.frame.listbox1 size]
        for {set ctr 0} {$ctr < $size} {incr ctr} {
            puts $fout [.icsum.frame.listbox1 get $ctr]
        }
        close $fout
    }
}

# Procedure: icsum_snap
proc icsum_snap {} {
    global icnewdesc
    global icselected
    global clockstr
    global datestr
    global loadname

# Update IC information
    set index $icselected
    incr index -1
    set lname [string range $loadname 0 11]
    set item [format " %s %s %s %-12s %s" $icselected $datestr $clockstr $lname $icnewdesc]
    .icsum.frame.listbox1 delete $index
    .icsum.frame.listbox1 insert $index $item

# Now save to file
    set fout [open ic/icsum.txt w]
    if {$fout != -1} {
        set size [.icsum.frame.listbox1 size]
        for {set ctr 0} {$ctr < $size} {incr ctr} {
            puts $fout [.icsum.frame.listbox1 get $ctr]
        }
        close $fout
    }
    css_cmd read "snap IC$icselected"
}

# Procedure: incr_index
proc incr_index {iname} {
    upvar $iname index
    if {[catch {incr index}]} {set index 1}
}

# Procedure: isdisplayed
proc isdisplayed {wname} {
# This procedure determines whether a window is visible and should be updated
#
# Returns: 1 if window exists and is visible
#          0 otherwise
    if {[wininfo exists $wname] && ([wm state $wname] == "normal")} {
        return 1
    } { return 0}
}

# Procedure: label_catch
proc label_catch {wname} {
    global dragbuffer
    global dragsrc

    if {$dragbuffer != ""} {
        set vname [lindex [$wname configure -textvariable] 4]
        if {$vname != ""} {
            eval "uplevel #0 {set $vname $dragbuffer}"
        }
        if {$dragsrc != "" && $dragsrc != $wname} {
            set vname [lindex [$dragsrc configure -textvariable] 4]
            if {$vname != ""} {
                eval "uplevel #0 {set $vname \"}"
            }
        }
        set dragbuffer ""
    }
    set dragsrc ""
}

```



```

}
}

# Procedure: label_drag
proc label_drag { wname screenX screenY } {
    global dragbuffer
    global dragsrc

    set vname [lindex [$wname configure -textvariable] 4]
    eval "uplevel #0 {set dragbuffer \$$vname}"
    set dragsrc $wname
    show.dragwin $screenX $screenY
}

```

```

# Procedure: label_setval
proc label_setval { wname } {
    $wname configure -borderwidth 0 -textvariable {}
    entry $wname.setval -width 13 -borderwidth 1 -relief sunken -fg black -bg lightsteelblue -takefocus 1 -textvariable icstemp2 -font
    -adobe-helvetica-medium-o-normal-10-100-75-75-p-57-iso8859-1
    pack configure $wname.setval
    focus $wname.setval
    bind $wname.setval <Key-Return> {
        set icstemp2 [string trimleft $icstemp2]
        set wparent [winfo parent %W]
        destroy %W
        set offset [string range [winfo name $wparent] 5 end]
        $wparent configure -borderwidth 2 -textvariable value($offset)
        if {$icstemp2 != ""} {
            incr offset $mps_base
            set vname [lindex $DataSrcList $offset]
            if {$vname != "a_z"} {
                css_cmd ctrl "setval $vname $icstemp2"
                mps_update
            }
        }
        set icstemp2 ""
    }
    bind $wname.setval <Key-Escape> {
        set wparent [winfo parent %W]
        destroy %W
        $wparent configure -borderwidth 2
        set icstemp2 ""
    }
}

```

```

# Procedure: label_setvar
proc label_setvar { wname } {
    $wname configure -borderwidth 0
    entry $wname.setvar -width 24 -borderwidth 1 -relief sunken -fg black -bg lightsteelblue -takefocus 1 -textvariable icstemp -font
    -adobe-helvetica-medium-o-normal-10-100-75-75-p-57-iso8859-1
    pack $wname.setvar
    focus $wname.setvar
    bind $wname.setvar <Key-Return> {
        set icstemp [string trimleft $icstemp]
        set wparent [winfo parent %W]
        set vname [lindex [$wparent configure -textvariable] 4]
        set $vname $icstemp
        destroy %W
        $wparent configure -borderwidth 2
        if {$icstemp != ""} {
            set offset [string range [winfo name $wparent] 4 end]
            incr offset $mps_base
            set DataSrcList [lreplace $DataSrcList $offset $offset $icstemp]
            mps_update
        }
        set icstemp ""
    }
    bind $wname.setvar <Key-Escape> {
        set wparent [winfo parent %W]
        destroy %W
        $wparent configure -borderwidth 2
        set icstemp ""
    }
}

```

```
}  
}
```

```
# Procedure: list_drag  
proc list_drag { wname ypos screenX screenY } {  
    global dragbuffer  
    global dragsrc  
    global qindex  
  
    set index [$wname nearest $ypos]  
    set line [$wname get $index]  
    set vname [lindex $line 0]  
    set ddim [lindex $line 1]  
    if {[string index $ddim 0] == "("} {  
        set ndim [length [split $ddim ,]]  
        set ddimstr ""  
        set sep "("  
        for {set ctr 1} {$ctr <= $ndim} {incr ctr} {  
            set ddimstr "$ddimstr$sep$qindex($ctr)"  
            set sep ","  
        }  
        set ddimstr "$ddimstr)"  
    } {set ddimstr ""}  
    set dragbuffer "$vname$ddimstr"  
    # set dragsrc ""  
    show.dragwin $screenX $screenY  
}
```

```
# Procedure: mfi_fill  
proc mfi_fill { filter } {  
    set fstr [string trim $filter]  
    if {$fstr == ""} {set fstr ""}  
    .mfi.frame.listbox1 delete 0 end  
    set mflist [css_cmd stat mfinfo]  
    foreach remfunc $mflist {  
        set mfname [lindex $remfunc 0]  
        set desc [lindex $remfunc end]  
        set line [format "%-16s $desc" $mfname $desc]  
        if {[regexp -nocase $fstr $line]} {.mfi.frame.listbox1 insert end $line}  
    }  
}
```

```
# Procedure: mfp  
proc mfp { mfname } {  
    global mfp_fname  
  
    ShowWindow.mfp  
    mfp_fill $mfname  
    mfp_getstatus $mfp_fname  
}
```

```
# Procedure: mfp_apply  
proc mfp_apply {} {  
    global mfststatus  
    global mfpdesc  
    global mfp_mfname  
    global mfp_fname  
    global mfststatus  
    global init_severity_req  
    global ramp_time_req  
    global final_severity_req  
    global delay_req  
    global trigger_req  
  
    if {$mfstatus == "OFF"} {  
        set initsev 0  
        set ramptime 0  
        set finsev 0  
        set delay 0  
        set trigger 0  
    }  
}
```

```

set initsev [string trim $init_severity_req]
if {$initsev == ""} {set initsev 1}
set ramp_time [string trim $ramp_time_req]
set finsev [string trim $final_severity_req]
if {$finsev == ""} {set finsev 1}
if {$ramp_time == ""} {set ramp_time 0}
if {$ramp_time == 0} {set finsev $initsev}
set delay [string trim $delay_req]
if {$delay == ""} {set delay 0}
set trigger [string trim $trigger_req]
}
mmf $mfp_mfname $mfp_fname $mfstatus $initsev $ramp_time $finsev $delay $trigger
}

```

```

# Procedure: mfp_fill
proc mfp_fill { mfname } {
global mfpdesc
global mfp_mfname
global mfp_fname
global mfstatus
global init_severity_req
global ramp_time_req
global final_severity_req
global delay_req
global trigger_req

if {[winfo exists .mfp]} {
set malf [lindex [css_cmd stat "mfinfo $mfname"] 0]
set mfp_mfname [lindex $malf 0]
set mfp_fname [lindex $malf 1]
set status [lindex $malf 2]
if {$status} {
set mfstatus Active
} {
set mfstatus OFF}
set size [lindex $malf 3]
set lastpos [expr $size - 2]
set range [lrange $malf 4 $lastpos]
set mfpdesc [lindex $malf end]
set init_severity_req ""
set ramp_time_req ""
set final_severity_req ""
set delay_req ""
set trigger_req ""
if {$mfp_mfname == $mfp_fname} {
set type discrete
.mfp.frame16.entry28 configure -bg systembuttonface -state disabled
.mfp.frame17.entry30 configure -bg systembuttonface -state disabled
.mfp.frame18.entry31 configure -bg systembuttonface -state disabled
} {
set type variable
.mfp.frame16.entry28 configure -bg azure -state normal
.mfp.frame17.entry30 configure -bg azure -state normal
.mfp.frame18.entry31 configure -bg azure -state normal
}
}
}
}

```

```

# Procedure: mfp_getstatus
proc mfp_getstatus { plid } {
global mfp_fname
global mfpfeedback

if {[winfo exists .mfp] && $plid == $mfp_fname} {
set mfpfeedback [css_cmd stat "getpos $plid"]
set cmd "mfp_getstatus $plid"
after 1000 $cmd
}
}

```

```

# Procedure: mfs_getstatus
proc mfs_getstatus {} {

```

```

if {[isdisplayed .mfs]} {
    set mfslist [css_cmd stat "mfstat all"]
    set orgsel [.mfs.frame.listbox1 curselection]
    set orgidx [.mfs.frame.listbox1 nearest 0]
    .mfs.frame.listbox1 delete 0 end
    eval ".mfs.frame.listbox1 insert end $mfslist"
    .mfs.frame.listbox1 yview $orgidx
    if {$orgsel != ""} {.mfs.frame.listbox1 selection set $orgsel}
}
}

# Procedure: mmf
proc mmf { name ffname state initsev ramptime finsev delay trigger } {
    css_cmd ctrl "mfmod $name $ffname $state $initsev $ramptime $finsev $delay $trigger"
}

# Procedure: move_dragwin
proc move_dragwin { screenX screenY } {
    # This procedure moves the dragbuffer window
    #
    wm geometry .dragwin +[expr $screenX - 10]+[expr $screenY - 10]
}

# Procedure: move_icon
proc move_icon {} {
    global icon_on

    if {$icon_on} {
        move_dragwin [winfo pointerx .] [winfo pointery .]
        after 10 move_icon
    }
}

# Procedure: mps_catch
proc mps_catch { wname } {
    global dragbuffer
    global dragsrc
    global mps_base
    global DataSrcList

    if {$dragbuffer != ""} {
        if {$dragsrc != ""} {
            set offset [string range [winfo name $dragsrc] 4 end]
            incr offset $mps_base
            set DataSrcList [replace $DataSrcList $offset $offset "a_z"]
        }
        set offset [string range [winfo name $wname] 4 end]
        incr offset $mps_base
        set DataSrcList [replace $DataSrcList $offset $offset $dragbuffer]
        label_catch $wname
        mps_update
    }
}

# Procedure: mps_cyclic
proc mps_cyclic {} {
    global mpslist
    global value
    global datalistid

    if {[winfo exists .monpam]} {
        set vlist [css_cmd stat "getlist $datalistid"]
        set ctr 0
        foreach item $mpslist {
            if {$item == "a_z"} {
                set value($ctr) ""
            }
            set value($ctr) [[index $vlist $ctr]]
        }
        incr ctr
    }
}

```

```

}
}

# Procedure: mps_insert
proc mps_insert {} {
  global dragbuffer
  global dragsrc
  global mps_base
  global DataSrcList

  set types {
    {"ICS monitored parameters"    { .mps } }
    {"Text files"                  { .txt } }
    {"All files"                    * }
  }

  set result [tk_getOpenFile -filetypes $types -initialdir ./mp -title "Insert Monitored Parameter List" -defaulttextextension .mps]
  if {$result != ""} {
    # Retrieve list to insert
    set fin [open $result r]
    gets $fin InsSrcList
    close $fin

    # Insert items into non-occupied slots
    set occupied ""
    set pos 0
    foreach item $InsSrcList {
      if {[index $DataSrcList $pos] == "a_z"} {
        set DataSrcList [replace $DataSrcList $pos $pos [index $InsSrcList $pos]]
        } { lappend occupied [index $InsSrcList $pos] }
        incr pos
      }

    # Insert remaining items into first available slots
    foreach item $occupied {
      set pos [search -exact $DataSrcList a_z]
      if {$pos != -1} {
        set DataSrcList [replace $DataSrcList $pos $pos $item]
      }
    }

    # Refresh the monitored parameter window
    mps_update
  }
}

```

```

# Procedure: mps_load
proc mps_load {} {
  global mps_height
  global mps_base
  global DataSrcList
  global DataValList
  global name
  global value
  global units
  global desc
  global datalistid

  set types {
    {"ICS monitored parameters"    { .mps } }
    {"Text files"                  { .txt } }
    {"All files"                    * }
  }

  set result [tk_getOpenFile -filetypes $types -initialdir ./mp -title "Load Monitored Parameter List" -defaulttextextension .mps]
  if {$result != ""} {
    set fin [open $result r]
    gets $fin DataSrcList
    close $fin
    mps_update
  }
}

```

```

# Procedure: mps_save
proc mps_save {} {
global DataSrcList

set types {
    {"ICS monitored parameters"    {.mps}    }
    {"Text files"                  {.txt}    }
    {"All files"                    ""}
}

set result [tk_getSaveFile -filetypes $types -initialdir ./mp -title "Save Monnitored Parameter List" -initialfile Untitled -defaultextension .mps]
if {$result != ""} {
    set fout [open $result w]
    puts $fout $DataSrcList
    close $fout
}
}

```

```

# Procedure: mps_scroll
proc mps_scroll { action args} {
global mps_base
global mps_maxvar
global mps_height

switch $action {
    scroll {
        set number [lindex $args 0]
        if {[lindex $args 1] == "pages"} {
            set delta $mps_height
        } {set delta 1}
        if {$number == 1} {
            set mps_base [expr $mps_base + $delta]
        } elseif {$number == -1} {
            set mps_base [expr $mps_base - $delta]
        }
    }
    moveto {
        set maxbase [expr $mps_maxvar - $mps_height]
        set mps_base [expr $args * $mps_maxvar]
    }
    default {puts $action}
}

set mps_base [expr int($mps_base)]
if {$mps_base < 0} {set mps_base 0}
set maxbase [expr $mps_maxvar - $mps_height]
if {$mps_base >= $maxbase} {set mps_base $maxbase}
set first [expr $mps_base * 1.0 / $mps_maxvar]
set last [expr ($mps_base + $mps_height) * 1.0 / $mps_maxvar]
.monparm.frame37.frame38.mpscroll set $first $last
after 100 {mps_update}
}

```

```

# Procedure: mps_update
proc mps_update {} {
global mps_height
global mps_base
global mpslist
global DataSrcList
global DataVallList
global name
global value
global units
global desc
global datalistid

if {[winfo exists .monparm]} {
    set last [expr $mps_base + $mps_height - 1]
    set mpslist [lrange $DataSrcList $mps_base $last]
    css_cmd stat "setlist $datalistid $mpslist"
    set vlist [css_cmd stat "getlist $datalistid"]
    set ctr 0
    foreach item $mpslist {
        if {$item == "a_z"} {
            set name($ctr) ""

```

```

        set value($ctr) ""
        set units($ctr) ""
        set desc($ctr) ""
    }{
        set name($ctr) $item
        set value($ctr) [index $vlist $ctr]
        set units($ctr) [css_cmd ctrl "ptid_info $item units"]
        set desc($ctr) [css_cmd ctrl "ptid_info $item briefdesc"]
    }
    }
    incr ctr
}
}
}

```

Procedure: read_ics

```

proc read_ics {} {
if {[winfo exists .icsum.frame.listbox1]} {
    set fin [open ic/icsum.txt r]
    if {$fin != -1} {
        .icsum.frame.listbox1 delete 0 end
        gets $fin line
        while (![eof $fin]) {
            set icno [index $line 0]
            set date [index $line 1]
            set time [index $line 2]
            set name [index $line 3]
            set desc [range $line 4 end]
            if {[string range $desc 0 0] == ""} {
                set dstat ""
            }{
                set dstat " "
            }
            set lname [string range $name 0 11]
            set item [format "%s %s %s %s %-12s %s%s" $icno $date $time $lname $dstat $desc]
            .icsum.frame.listbox1 insert end $item
            gets $fin line
        }
        close $fin
    }
}
}
}

```

Procedure: rfi_fill

```

proc rfi_fill { filter } {
set fstr [string trim $filter]
if {$fstr == ""} {set fstr " "}
.rfi.frame.listbox1 delete 0 end
set rfiist [css_cmd stat rfiinfo]
foreach remfunc $rfiist {
    set rfname [index $remfunc 0]
    set status [index $remfunc 2]
    set size [length $remfunc]
    set lastpos [expr $size - 2]
    set range [range $remfunc 4 $lastpos]
    set desc [index $remfunc end]
    set line [format "%-16s %8s %16s $desc" $rfname $status $range]
    if {[regexp -nocase $fstr $line]} { .rfi.frame.listbox1 insert end $line }
}
}
}

```

Procedure: rfp

```

proc rfp { rfname } {
global rfp_fname

ShowWindow.rfp
rfp_fill $rfname
rfp_getstatus $rfp_fname
}

```

Procedure: rfp_fill

```

proc rfp_fill { rfname } {
global rfp_desc
global rfp_min

```

```

global rfpmax
global rfpjoin
global rfp_rfname
global rfp_fbnme
global rfp_requested

if {[winfo exists .rfp]} {
    .rfp.frame2.frame.listbox1 delete 0 end
    set remfunc [lindex [css_cmd stat "rinfo $rfname"] 0]
    set rfp_rfname [lindex $remfunc 0]
    set rfp_fbnme [lindex $remfunc 1]
    set status [lindex $remfunc 2]
    set type [lindex $remfunc 3]
    set size [lindex $remfunc 4]
    set lastpos [expr $size - 2]
    set range [lrange $remfunc 4 $lastpos]
    set rfpdesc [lindex $remfunc end]
    set rfp_requested ""
    if {$type == 2} {
        set rfpmin [lindex $range 0]
        set rfpjoin to
        set rfpmax [lindex $range 1]
        .rfp.frame2.frame3.frame5.entry10 configure -state normal
    }
    .rfp.frame2.frame3.frame5.entry10 configure -state disabled
    set rfpmin ""
    set rfpjoin ""
    set rfpmax ""
    eval ".rfp.frame2.frame.listbox1 insert end $range"
}
}
}

```

```

# Procedure: rfp_getstatus
proc rfp_getstatus { ptid } {
    global rfp_fbnme
    global rfpfeedback

    if {[winfo exists .rfp] && $ptid == $rfp_fbnme} {
        set rfpfeedback [css_cmd stat "getpos $ptid"]
        set cmd "rfp_getstatus $ptid"
        after 1000 $cmd
    }
}

```

```

# Procedure: rfp_set
proc rfp_set { value } {
    global rfp_rfname
    global rfp_requested
    global rfpmin
    global rfpmax

    if {$rfpmin == ""} {
        set rfp_requested $value
        css_cmd ctrl "setpos $rfp_rfname $value"
    }
    if {($value >= $rfpmin) && ($value <= $rfpmax)} {
        css_cmd ctrl "setval $rfp_rfname $value"
    }
}
}

```

```

# Procedure: sendmsg
proc sendmsg { cmdstr } {
    global datasrc

    puts $datasrc "eval"
    puts $datasrc $cmdstr
    flush $datasrc
    gets $datasrc result
    puts $result
}

```



```

# Procedure: show_dragwin
proc show_dragwin { screenX screenY } {
# This procedure creates a top level window which displays the
# drag buffer.
#
global dragbuffer

if {[winfo exists .dragwin]} {return}
toplevel .dragwin
label .dragwin.buffer -highlightthickness 0 -relief raised -borderwidth 1 -fg white -bg steelblue -text $dragbuffer -padx 0 -pady 0
pack .dragwin.buffer
wm overriddenirect .dragwin 1
wm geometry .dragwin +[expr $screenX - 10]+[expr $screenY - 10]
}

```

```

# Procedure: showdiag
proc showdiag { filename } {
    css_cmd ctrl "launch $filename [info hostname]"
}

```

```

# Procedure: vcs_diag
proc vcs_diag { filename host } {
global machine

```

```

set orghost $machine
switchdest $host
css_cmd ctrl "launch $filename $orghost"
switchdest $orghost
}

```

```

# Procedure: vcs_getstatus
proc vcs_getstatus {} {
# Call to update VCS
#
global hostnames
global vcsdiag
global vcs_passno

```

```

if {[winfo exists .vcs]} {return}
set newlist [css_cmd stat getservers]
if {$newlist != $hostnames} {
    vcs_show
    set hostnames $newlist
}
foreach item $hostnames {
    set dlist [css_cmd stat "getdiag $item"]
    if {$dlist == ""} {
        set vcsdiag($item) ""
    } {
        set totdiags [length $dlist]
        set pos [expr $vcs_passno % $totdiags]
        set vcsdiag($item) [index $dlist $pos]
    }
}
incr vcs_passno
}

```

```

# Procedure: vcs_kill
proc vcs_kill { filename host } {
css_cmd ctrl "kill ${host};$filename"
}

```

```

# Procedure: vcs_show
proc vcs_show {} {
# This procedure constructs a .vcs toplevel window based on
# the list of hosts given in hostnames
#
global vcsdiag

```

```

global hostnames

if {[winfo exists .vcs]} {
    wm deiconify .vcs
    if {[info commands XFDestroy] != ""} {
        catch {XFDestroy .vcs.baseframe}
        { catch "destroy .vcs.baseframe" }
    }
    {
        toplevel .vcs
        frame .vcs.buttons
        pack .vcs.buttons -fill x -side bottom
        button .vcs.buttons.done -bg slategrey -fg green -text Done -command {
            if {[info commands XFDestroy] != ""} {
                catch {XFDestroy .vcs}
                { catch "destroy .vcs" }
            }
        }
    }
    -font -adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1
    pack .vcs.buttons.done -fill x
    label .vcs.trash -image trashcan -bg grey70 -relief raised -borderwidth 2
    set cmd "bind .vcs.trash <Enter> {if {$dragbuffer != \v} {vcs_kill $dragbuffer $dragbuffer2}}"
    eval $cmd
    pack .vcs.trash -fill x -side bottom
}

frame .vcs.baseframe
pack .vcs.baseframe -side top
set framectr 0
set hctr 3
set hostnames [css_cmd stat getservers]
foreach host $hostnames {
    incr hctr
    if {$hctr > 3} {
        set framename ".vcs.baseframe.frame$framectr"
        frame $framename
        pack configure $framename -side top
        incr framectr
        set hctr 1
    }
    frame $framename.$host
    pack configure $framename.$host -side left

    label $framename.$host.name -relief raised -borderwidth 2 -fg white -bg steelblue -text $host -width 8
    pack configure $framename.$host.name -side top

    label $framename.$host.target -relief raised -borderwidth 2 -bg lightsteelblue -width 8 -height 3 -textvariable vcsdiag($host)
    set cmd "bind $framename.$host.target <Enter> {if {$dragbuffer != \v} {vcs_diag $dragbuffer $host}}"
    eval $cmd
    eval "bind $framename.$host.target <Button-1> {set dragbuffer \vcsdiag($host); set dragbuffer2 $host; show.dragwin %X %Y}"
    eval "bind $framename.$host.target <B1-Motion> {move.dragwin %X %Y}"
    eval "bind $framename.$host.target <ButtonRelease-1> {drop.dragwin}"
    pack configure $framename.$host.target -side top
}
while {$hctr < 3} {
    frame $framename.blank$hctr
    pack configure $framename.blank$hctr -side left

    label $framename.blank$hctr.name -relief raised -borderwidth 2 -bg steelblue -text "" -width 8
    pack configure $framename.blank$hctr.name -side top

    label $framename.blank$hctr.target -relief raised -borderwidth 2 -bg lightsteelblue -width 8 -height 3 -text N/A
    pack configure $framename.blank$hctr.target -side left
    incr hctr
}
}

# Procedure: vcs_trash
proc vcs_trash {} {
    global machine
    global dragbuffer
    global dragbuffer2

    set orghost $machine
    set host $dragbuffer2
    if {$host != ""} {
        switchdest $host
        set diag $dragbuffer
    }
}

```

```

if {$diag != ""} {css_cmd ctrl "kill ${host}:$diag"}
switchdest $orghost
}
}

```

User defined images

```

image create photo trashcan \
-file {trash.gif} \
-gamma {1.0} \
-height {0} \
-width {0}

```

Internal procedures

```
# Procedure: Alias
```

```
if {"[info procs Alias]" == ""} {
```

```
proc Alias { args } {
```

```
# xf ignore me 7
```

```
#####
```

```
# Procedure: Alias
```

```
# Description: establish an alias for a procedure
```

```
# Arguments: args - no argument means that a list of all aliases
```

```
# is returned. Otherwise the first parameter is
```

```
# the alias name, and the second parameter is
```

```
# the procedure that is aliased.
```

```
# Returns: nothing, the command that is bound to the alias or a
```

```
# list of all aliases - command pairs.
```

```
# Sideeffects: internalAliasList is updated, and the alias
```

```
# proc is inserted
```

```
#####
```

```
global internalAliasList
```

```
if {[length $args] == 0} {
```

```
return $internalAliasList
```

```
}
```

```
if {[length $args] == 1} {
```

```
set xTmpIndex [lsearch $internalAliasList "[index $args 0]"]
```

```
if {$xTmpIndex != -1} {
```

```
return [index [index $internalAliasList $xTmpIndex] 1]
```

```
}
```

```
}}
```

```
if {[length $args] == 2} {
```

```
eval "proc [index $args 0] {args} {#xf ignore me 4
```

```
return [eval "[index $args 1] \${args}"]"
```

```
set xTmpIndex [lsearch $internalAliasList "[index $args 0]"]
```

```
if {$xTmpIndex != -1} {
```

```
set internalAliasList [replace $internalAliasList $xTmpIndex $xTmpIndex "[index $args 0] [index $args 1]"]
```

```
}
```

```
lappend internalAliasList "[index $args 0] [index $args 1]"
```

```
}
```

```
}}
```

```
error "Alias: wrong number or args: $args"
```

```
}
```

```
}
```

```
}
```

```
}
```

```
# Procedure: Cat
```

```
if {"[info procs Cat]" == ""} {
```

```
proc Cat { filename } {
```

```
# xf ignore me 7
```

```
#####
```

```
# Procedure: Cat
```

```
# Description: emulate UNIX cat for one file
```

```
# Arguments: filename
```

```
# Returns: file contents
```

```
# Sideeffects: none
```

```
#####
```

```
global tcl_platform
```

```

if {$tcl_platform(platform) == "unix"} {exec cat $filename} {
  set fileid [open $filename "r"]
  set data [read $fileid]
  close $fileid
  return $data
}
}

# Procedure: Chmod
if {"[info procs Chmod]" == ""} {
proc Chmod { mode file} {
# xf ignore me 7
#####
# Procedure: Chmod
# Description: ignore UNIX chmod under DOS
# Arguments: file1 file2/directory
# Returns: nothing
# Sideeffects: none
#####
global tcl_platform
if {$tcl_platform(platform) == "unix"} {eval exec chmod $mode $file} {
  regsub -all {} $file1 {\} file1
  regsub -all {} $file2 {\} file2
  eval exec command.com /c copy /y $file1 $file2 >@stderr
}
}

# Procedure: Cp
if {"[info procs Cp]" == ""} {
proc Cp { file1 file2} {
# xf ignore me 7
#####
# Procedure: Cp
# Description: emulate UNIX cp with DOS COPY
# Arguments: file1 file2/directory
# Returns: nothing
# Sideeffects: none
#####
global tcl_platform
if {$tcl_platform(platform) == "unix"} {eval exec cp $file1 $file2} {
  regsub -all {} $file1 {\} file1
  regsub -all {} $file2 {\} file2
  eval exec command.com /c copy /y $file1 $file2 >@stderr
}
}

# Procedure: GetSelection
if {"[info procs GetSelection]" == ""} {
proc GetSelection {} {
# xf ignore me 7
#####
# Procedure: GetSelection
# Description: get current selection
# Arguments: none
# Returns: none
# Sideeffects: none
#####

# the save way
set xfSelection ""
catch "selection get" xfSelection
if {"$xfSelection" == "selection doesn't exist or form \"STRING\" not defined"} {
  return ""
}
return $xfSelection
}
}
}

```

```

# Procedure: Ls
if {[info procs Ls] == ""} {
proc Ls { args} {
# xf ignore me 7
#####
# Procedure: Ls
# Description: emulate UNIX ls
# Arguments: like UNIX (switches authorized -F -a [-a is ignored])
# Returns: directory list
# Sideeffects: none
#####
global tcl_platform
if {$tcl_platform(platform) == "unix"} {eval exec ls $args} {
set last [lindex $args end]
if {[lsearch $last -" ] >= 0 || $last == ""} {set path ""} {set path $last/*}
set lst [glob $path]
set lst1 ""
if {[lsearch -exact $args "-F"] >= 0} then {set car "F"} else {set car ""}
foreach f $lst {
if {[file isdirectory $f]} {set f [file tail $f]$car} {set f [file tail $f]}
lappend lst1 $f
}
return [lsort $lst1]
}
}
}

```

```

# Procedure: NoFunction
if {[info procs NoFunction] == ""} {
proc NoFunction { args} {
# xf ignore me 7
#####
# Procedure: NoFunction
# Description: do nothing (especially with scales and scrollbars)
# Arguments: args - a number of ignored parameters
# Returns: none
# Sideeffects: none
#####
}
}

```

```

# Procedure: Rm
if {[info procs Rm] == ""} {
proc Rm { filename} {
# xf ignore me 7
#####
# Procedure: Rm
# Description: emulate UNIX rm with DOS DEL
# Arguments: filename(s)
# Returns: nothing
# Sideeffects: none
#####
global tcl_platform
if {$tcl_platform(platform) == "unix"} {exec rm -f $filename} {
regsub -all {} $filename {} filename
eval exec command.com /c del $filename >@stderr
}
}
}

```

```

# Procedure: SN
if {[info procs SN] == ""} {
proc SN { {xfName ""}} {
# xf ignore me 7
#####
# Procedure: SN
# Description: map a symbolic name to the widget path
# Arguments: xfName
# Returns: the symbolic name
# Sideeffects: none
#####
}
}

```

```

SymbolicName $xfName
}
}

```

```

# Procedure: SymbolicName
if {"[info procs SymbolicName]" == ""} {
proc SymbolicName { (xfName "") } {
# xf ignore me 7
#####
# Procedure: SymbolicName
# Description: map a symbolic name to the widget path
# Arguments: xfName
# Returns: the symbolic name
# Sideeffects: none
#####

```

```

global symbolicName

if {"$xfName" != ""} {
set xfArrayName ""
append xfArrayName symbolicName ( $xfName )
if {[catch "set \"$xfArrayName\" xfValue"]} {
return $xfValue
}
if {"[info commands XFProcError]" != ""} {
XFProcError "Unknown symbolic name:\n$xfName"
}
puts stderr "XF error: unknown symbolic name:\n$xfName"
}
}
return ""
}
}

```

```

# Procedure: Unalias
if {"[info procs Unalias]" == ""} {
proc Unalias { aliasName } {
# xf ignore me 7
#####
# Procedure: Unalias
# Description: remove an alias for a procedure
# Arguments: aliasName - the alias name to remove
# Returns: none
# Sideeffects: internalAliasList is updated, and the alias
#             proc is removed
#####

```

```

global internalAliasList

set xfIndex [lsearch $internalAliasList "$aliasName "]
if {$xfIndex != -1} {
rename $aliasName ""
set internalAliasList [lreplace $internalAliasList $xfIndex $xfIndex]
}
}
}

```

```

# application parsing procedure
proc XFLocalParseAppDefs (xfAppDefFile) {
global xfAppDefaults tcl_platform

```

```

# basically from: Michael Moore
if {[file exists $xfAppDefFile] &&
[file readable $xfAppDefFile] &&
[file type $xfAppDefFile] == "link"} {
catch "file type $xfAppDefFile" xfType
while {"$xfType" == "link"} {
if {[catch "file readlink $xfAppDefFile" xfAppDefFile]} {
return
}
}
}

```

```

    catch "file type $xfAppDefFile" xfType
}
}
if {!("$xfAppDefFile" != "" &&
    [file exists $xfAppDefFile] &&
    [file readable $xfAppDefFile] &&
    "file type $xfAppDefFile" == "file")} {
    return
}
if {[!catch "open $xfAppDefFile r" xfResult]} {
    set xfAppFileContents [read $xfResult]
    close $xfResult
    foreach line [split $xfAppFileContents "\n"] {
        # backup indicates how far to backup. It applies to the
        # situation where a resource name ends in . and when it
        # ends in *. In the second case you want to keep the *
        # in the widget name for pattern matching, but you want
        # to get rid of the . if it is the end of the name.
        set backup -2
        set line [string trim $line]
        if {[string index $line 0] == "#" || "$line" == ""} {
            # skip comments and empty lines
            continue
        }
        if {[string compare "windows" $tcl_platform(platform)]} {
            set list [split $line ";"]
        } {
            set list [split $line "."]
        }
        set resource [string trim [lindex $list 0]]
        set i [string last "." $resource]
        set j [string last "*" $resource]
        if {$j > $i} {
            set i $j
            set backup -1
        }
        incr i
        set name [string range $resource $i end]
        incr i $backup
        set widname [string range $resource 0 $i]
        set value [string trim [lindex $list 1]]
        if {"$widname" != "" && "$widname" != ""} {
            # insert the widget and resourcename to the application
            # defaults list.
            if {[!info exists xfAppDefaults]} {
                set xfAppDefaults ""
            }
            lappend xfAppDefaults [list $widname [string tolower $name] $value]
        }
    }
}
}
}

# application loading procedure
proc XFLocalLoadAppDefs {{xfClasses ""} {xfPriority "startupFile"} {xfAppDefFile ""}} {
    global env tcl_platform

    if {[!string compare "windows" $tcl_platform(platform)]} {
        set separator ";"
    } {
        set separator "."
    }
}
if {"$xfAppDefFile" == ""} {
    set xfFileList ""
    if {[!info exists env(XUSERFILESEARCHPATH)]} {
        eval lappend xfFileList [split $env(XUSERFILESEARCHPATH) $separator]
    }
    if {[!info exists env(XAPPLRESDIR)]} {
        eval lappend xfFileList [split $env(XAPPLRESDIR) $separator]
    }
    if {[!info exists env(XFILESEARCHPATH)]} {
        eval lappend xfFileList [split $env(XFILESEARCHPATH) $separator]
    }
}
append xfFileList "/usr/lib/X11/app-defaults"
append xfFileList "/usr/X11/lib/X11/app-defaults"

```



```

set {desc(1)} {}
set {desc(10)} {}
set {desc(11)} {}
set {desc(12)} {}
set {desc(13)} {}
set {desc(14)} {}
set {desc(15)} {}
set {desc(16)} {}
set {desc(17)} {}
set {desc(18)} {}
set {desc(19)} {}
set {desc(2)} {}
set {desc(20)} {}
set {desc(21)} {}
set {desc(22)} {}
set {desc(23)} {}
set {desc(24)} {}
set {desc(25)} {}
set {desc(26)} {}
set {desc(27)} {}
set {desc(28)} {}
set {desc(29)} {}
set {desc(3)} {}
set {desc(4)} {}
set {desc(5)} {}
set {desc(6)} {}
set {desc(7)} {}
set {desc(8)} {}
set {desc(9)} {}
global {dragbuffer}
set {dragbuffer} {}
global {dragbuffer2}
set {dragbuffer2} {pentium}
global {dragsrc}
set {dragsrc} {}
global {ei}
set {ei(1)} {0}
set {ei(10)} {0}
set {ei(2)} {0}
set {ei(3)} {0}
set {ei(4)} {0}
set {ei(5)} {0}
set {ei(6)} {0}
set {ei(7)} {0}
set {ei(8)} {0}
set {ei(9)} {0}
global {ei_ctrl}
set {ei_ctrl(1)} {OFF}
set {ei_ctrl(10)} {OFF}
set {ei_ctrl(2)} {OFF}
set {ei_ctrl(3)} {OFF}
set {ei_ctrl(4)} {OFF}
set {ei_ctrl(5)} {OFF}
set {ei_ctrl(6)} {OFF}
set {ei_ctrl(7)} {OFF}
set {ei_ctrl(8)} {OFF}
set {ei_ctrl(9)} {OFF}
global {ei_desc}
set {ei_desc(1)} {}
set {ei_desc(10)} {}
set {ei_desc(2)} {}
set {ei_desc(3)} {}
set {ei_desc(4)} {}
set {ei_desc(5)} {}
set {ei_desc(6)} {}
set {ei_desc(7)} {}
set {ei_desc(8)} {}
set {ei_desc(9)} {}
global {ei_fname}
set {ei_fname(1)} {}
set {ei_fname(10)} {}
set {ei_fname(2)} {}
set {ei_fname(3)} {}
set {ei_fname(4)} {}
set {ei_fname(5)} {}

```

```
set {et_fname(6)} {}
set {et_fname(7)} {}
set {et_fname(8)} {}
set {et_fname(9)} {}
global {final_severity_req}
set {final_severity_req} {}
global {firs}
set {firs} {0.0}
global {flis}
set {flis} {ed1.tcl ed2.tcl}
global {headingb1}
set {headingb1} {slategrey}
global {hostnames}
set {hostnames} {asus}
global {icnewdesc}
set {icnewdesc} {Electrical Network}
global {icno}
set {icno} {02}
global {icon_on}
set {icon_on} {0}
global {ics_alarm}
set {ics_alarm} {off}
global {ics_alarm_new}
set {ics_alarm_new} {off}
global {ics_confirm}
set {ics_confirm} {off}
global {ics_confirm_new}
set {ics_confirm_new} {off}
global {ics_malfstat}
set {ics_malfstat} {malf: 0:0}
global {ics_pref_new}
set {ics_pref_new} {off}
global {ics_restrict}
set {ics_restrict} {off}
global {ics_restrict_new}
set {ics_restrict_new} {off}
global {icsselected}
set {icsselected} {50}
global {icstemp}
set {icstemp} {}
global {icstemp2}
set {icstemp2} {}
global {index}
set {index} {1}
global {init_severity}
set {init_severity} {entry28}
global {init_severity_req}
set {init_severity_req} {}
global {item}
set {item} { 0 0 0 0 0}
global {last}
set {last} {0.1}
global {line}
set {line} { 50 08/24/97 13:01:19 RTSCHED.EXE *Electrical Network}
global {list_id}
set {list_id(asus)} {17}
global {list_id2}
set {list_id2(asus)} {18}
global {lnam}
set {lnam} {rtsched}
global {loadname}
set {loadname} {rtsched.exe}
global {machine}
set {machine} {asus}
global {menuTearoff}
set {menuTearoff} {0}
global {mfp_fbnam}
set {mfp_fbnam} {edvreg_fail_fb}
global {mfp_mfnam}
set {mfp_mfnam} {edvreg_fail}
global {mfpdesc}
set {mfpdesc} {Voltage regulator setpt failure}
global {mfpfeedback}
set {mfpfeedback} {1}
global {mfstatus}
```



```
set {rfpjoin} {}
global {rfpmax}
set {rfpmax} {}
global {rfpmin}
set {rfpmin} {}
global {rfpname}
set {rfpname} {ypriedhtr1switch}
global {selmachine}
set {selmachine} {cyrix}
global {simsecs}
set {simsecs} {5853}
global {simtime}
set {simtime} {01:37:33}
global {standalone}
set {standalone} {0}
global {stathost}
set {stathost} {Connections from asus}
global {status}
set {status} {Run}
global {time}
set {time} {08:00:00}
global {trigger_req}
set {trigger_req} {}
global {units}
set {units(0)} {}
set {units(1)} {}
set {units(10)} {}
set {units(11)} {}
set {units(12)} {}
set {units(13)} {}
set {units(14)} {}
set {units(15)} {}
set {units(16)} {}
set {units(17)} {}
set {units(18)} {}
set {units(19)} {}
set {units(2)} {}
set {units(20)} {}
set {units(21)} {}
set {units(22)} {}
set {units(23)} {}
set {units(24)} {}
set {units(25)} {}
set {units(26)} {}
set {units(27)} {}
set {units(28)} {}
set {units(29)} {}
set {units(3)} {}
set {units(4)} {}
set {units(5)} {}
set {units(6)} {}
set {units(7)} {}
set {units(8)} {}
set {units(9)} {}
global {value}
set {value(0)} {}
set {value(1)} {}
set {value(10)} {}
set {value(11)} {}
set {value(12)} {}
set {value(13)} {}
set {value(14)} {}
set {value(15)} {}
set {value(16)} {}
set {value(17)} {}
set {value(18)} {}
set {value(19)} {}
set {value(2)} {}
set {value(20)} {}
set {value(21)} {}
set {value(22)} {}
set {value(23)} {}
set {value(24)} {}
set {value(25)} {}
set {value(26)} {}
```

```

set {value(27)} {}
set {value(28)} {}
set {value(29)} {}
set {value(3)} {}
set {value(4)} {}
set {value(5)} {}
set {value(6)} {}
set {value(7)} {}
set {value(8)} {}
set {value(9)} {}
global {vcs_passno}
set {vcs_passno} {8459}
global {vcsdiag}
set {vcsdiag(asus)} {}
set {vcsdiag(pentium)} {}
set {vcsdiag(zenon)} {}
global {vname}
set {vname} {name(1)}
global {worldtime}
set {worldtime} {label26}
global {wparent}
set {wparent} {,monparm.frame37.frame38.name1}
global {xtemp1}
set {xtemp1} {948}
global {xtemp2}
set {xtemp2} {}
global {yesNoBox}
set {yesNoBox(activeBackground)} {}
set {yesNoBox(activeForeground)} {}
set {yesNoBox(afterNo)} {}
set {yesNoBox(afterYes)} {}
set {yesNoBox(anchor)} {}
set {yesNoBox(background)} {}
set {yesNoBox(button)} {}
set {yesNoBox(font)} {-abobe-helvetica-bold-r-normal-24-240-75-75-p-138-iso8859-1}
set {yesNoBox(foreground)} {}
set {yesNoBox(justify)} {center}

# please don't modify the following
# variables. They are needed by xf.
global {autoLoadList}
set {autoLoadList(ics.tcl)} {}
set {autoLoadList(main.tcl)} {}
global {internalAliasList}
set {internalAliasList} {}
global {moduleList}
set {moduleList(ics.tcl)} {}
global {preloadList}
set {preloadList(xfInternal)} {}
global {symbolicName}
set {symbolicName(root)} {}
global {xfWmSetPosition}
set {xfWmSetPosition} {,monparm.aps.pref}
global {xfWmSetSize}
set {xfWmSetSize} {,monparm.dispstat.pref}
global {xfAppDefToplevels}
set {xfAppDefToplevels} {}
}

# initialize global variables
InitGlobals

# display/remove toplevel windows.
ShowWindow.

global xfShowWindow.alarm
set xfShowWindow.alarm 0

global xfShowWindow.appmsg
set xfShowWindow.appmsg 0

global xfShowWindow.aps
set xfShowWindow.aps 0

global xfShowWindow.backtrack

```



```

set xfShowWindow.backtrack 0

global xfShowWindow.diagidx
set xfShowWindow.diagidx 0

global xfShowWindow.dispstat
set xfShowWindow.dispstat 0

global xfShowWindow.ets
set xfShowWindow.ets 0

global xfShowWindow.evs
set xfShowWindow.evs 0

global xfShowWindow.icsum
set xfShowWindow.icsum 0

global xfShowWindow.mfi
set xfShowWindow.mfi 0

global xfShowWindow.mfp
set xfShowWindow.mfp 0

global xfShowWindow.mfs
set xfShowWindow.mfs 0

global xfShowWindow.monparm
set xfShowWindow.monparm 0

global xfShowWindow.pref
set xfShowWindow.pref 0

global xfShowWindow.query
set xfShowWindow.query 0

global xfShowWindow.rfi
set xfShowWindow.rfi 0

global xfShowWindow.rfp
set xfShowWindow.rfp 0

# load default bindings.
if {[info exists env(XF_BIND_FILE)] &&
    "[info procs XFShowHelp]" == ""} {
    source $env(XF_BIND_FILE)
}

# initialize bindings for all widgets.
XFInitAllBindings

# parse and apply application defaults.
XFLocalLoadAppDefs lcs
XFLocalSetAppDefs

# end source
EndSrc

# eof
#

#!wish -f
# Program: icserver
# Tcl version: 7.6 (Tcl/Tk/XF)
# Tk version: 4.2
# XF version: 4.0
#
# _____
# Copyright (C) 1996 Dennis R. LaBelle (drlabel@albany.net) All Rights Reserved.
#
# PERMISSION IS GRANTED TO DISTRIBUTE THIS SOFTWARE FREELY, WITH THE
# EXCEPTION THAT ONE MAY NOT CHARGE FOR IT OR INCLUDE IT WITH SOFTWARE
# WHICH IS SOLD. Permission to use, copy, modify, and distribute this software and
# its documentation for educational, research, internal corporate and non-profit
# purposes, without fee, and without a written agreement is hereby granted for all

```

```

# cases that do not conflict with the restriction in the first sentence of this
# paragraph, provided that the above copyright notice, this paragraph, and the
# following three paragraphs appear in all copies.
#
# Permission to incorporate this software into commercial products may be obtained
# from the author (e-mail drtabel@albany.net).
#
# IN NO EVENT SHALL THE AUTHOR BE LIABLE TO ANY PARTY FOR DIRECT,
# INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF
# THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE AUTHOR HAS
# BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
# THE AUTHOR SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
# PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS"
# BASIS, AND THE AUTHOR HAS NO OBLIGATION TO PROVIDE MAINTENANCE,
# SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
# _____

```

```

# module inclusion
global env
global xflLoadPath
global xflLoadInfo
set xflMisc(separator) @
set xflLoadInfo 0
if {[info exists env(XF_LOAD_PATH)]} {
  if {[string first $env(XF_LOAD_PATH) @/lib] == -1} {
    set xflLoadPath $env(XF_LOAD_PATH):@/lib
  }
  set xflLoadPath @/lib
}
set xflLoadPath @/lib
}

global argc
global argv
set impArgv ""
for {set counter 0} {$counter < $argc} {incr counter 1} {
  case [string tolower [lindex $argv $counter]] in {
    {-xloadpath} {
      incr counter 1
      set xflLoadPath "[lindex $argv $counter]:$xflLoadPath"
    }
    {-xstartup} {
      incr counter 1
      source [lindex $argv $counter]
    }
    {-xbindfile} {
      incr counter 1
      set env(XF_BIND_FILE) "[lindex $argv $counter]"
    }
    {-xcolorfile} {
      incr counter 1
      set env(XF_COLOR_FILE) "[lindex $argv $counter]"
    }
    {-xcursorfile} {
      incr counter 1
      set env(XF_CURSOR_FILE) "[lindex $argv $counter]"
    }
    {-xfontfile} {
      incr counter 1
      set env(XF_FONT_FILE) "[lindex $argv $counter]"
    }
    {-xfmodelmono} {
      tk colormodel . monochrome
    }
    {-xfmodelcolor} {
      tk colormodel . color
    }
    {-xflloading} {
      set xflLoadInfo 1
    }
    {-xfnloading} {

```

```

    set xflLoadInfo 0
  }
  {default} {
    lappend tmpArgv [index $argv $counter]
  }
}
set argv $tmpArgv
set argc [llength $tmpArgv]
unset counter
unset tmpArgv

```

```

# procedure to show window .
proc ShowWindow. {args} {# xf ignore me 7

```

```

# Window manager configurations
wm positionfrom . user
wm sizefrom . ""
update idletask
wm iconify .
wm maxsize . 1152 845
wm minsize . 190 150
wm protocol . WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title . {xfmain.tcl on port 5000 connected to: s:\demo\rtsched.exe}

```

```

# build widget .frame1
frame .frame1

```

```

# build widget .frame1.button1
button .frame1.button1 \
  -borderwidth {1} \
  -command {quit} \
  -font {-adobe-helvetica-medium-o-normal-14-140-75-75-p-78-iso8859-1} \
  -padx {8} \
  -pady {4} \
  -text {Exit}

```

```

# build widget .frame1.button0
button .frame1.button0 \
  -borderwidth {1} \
  -command {.frame.text2 delete 1.0 end} \
  -font {-adobe-helvetica-medium-o-normal-14-140-75-75-p-78-iso8859-1} \
  -padx {8} \
  -pady {4} \
  -text {Clear Messages}

```

```

# build widget .frame
frame .frame \
  -relief {raised}

```

```

# build widget .frame.scrollbar1
scrollbar .frame.scrollbar1 \
  -borderwidth {1} \
  -command {.frame.text2 yview} \
  -highlightthickness {0} \
  -relief {raised}

```

```

# build widget .frame.scrollbar2
scrollbar .frame.scrollbar2 \
  -activerelief {sunken} \
  -borderwidth {1} \
  -command {.frame.text2 xview} \
  -highlightthickness {0} \
  -orient {horizontal} \
  -relief {raised}

```

```

# build widget .frame.text2
text .frame.text2 \
  -background {#c8dcdc} \
  -borderwidth {1} \
  -font {-adobe-helvetica-bold-o-normal-12-120-75-75-p-69-iso8859-1} \
  -height {6} \

```

```
-width {50} \  
-wrap {none} \  
-xscrollcommand {frame.scrollbar2 set} \  
-yscrollcommand {frame.scrollbar1 set}
```

```
# pack master .frame1  
pack configure .frame1.button0 \  
-fill y \  
-side left  
pack configure .frame1.button1 \  
-fill x \  
-side left
```

```
# pack master .frame  
pack configure .frame.scrollbar1 \  
-fill y \  
-side right  
pack configure .frame.scrollbar2 \  
-fill x \  
-side bottom  
pack configure .frame.text2 \  
-expand 1 \  
-fill both
```

```
# pack master .  
pack configure .frame1 \  
-anchor e \  
-side bottom  
pack configure .frame \  
-expand 1 \  
-fill both
```

```
.frame.text2 insert end {gelist 2  
status  
gelist 2  
status  
gelist 2  
status  
gelist 2  
status  
gelist 2  
status  
gelist 2  
status  
gelist 2  
status  
newlist 1  
newlist 2  
disconnect asus:xfmain.tcl}
```

```
if {[info procs XFEdit] != ""} {  
  catch "XFMiscBindWidgetTree ."  
  after 2 "catch {XFEditSetShowWindows}"  
}  
}
```

```
# User defined procedures
```

```
# Procedure: css_stat  
proc css_stat { cmd } {  
  css_cmd stat $cmd  
}
```

```
# Procedure: cyclic_server  
proc cyclic_server {} {  
  global cyclicrate  
  global statusinfo  
  global statstr  
  global cyclic_on  
  global bt_interval  
  global bt_next  
  global bt_curic
```

```
global bt_totbc
global bt_info
global load_name
global simtime
global malfstst
```

```
if {$cyclic_on} {
    putdata getstat
    getdata statusinfo
    set statstr $statusinfo
    getdata line
    lappend statusinfo $line
    getdata simtime
    lappend statusinfo $simtime
    getdata seconds
    lappend statusinfo $seconds
    getdata line
    lappend statusinfo $line
    getdata line
    lappend statusinfo $line
    getdata line
    lappend statusinfo $line
    lappend statusinfo $bt_interval
    lappend statusinfo $malfstst

    switch $statstr {
        Run {
            if {$seconds >= $bt_next} {
                set bt_next [expr $seconds + $bt_interval]
                putdata "snap BC$bt_curic"
                set bt_info($bt_curic) [format "%02d [clock format [clock seconds] -format "%D %T"] $simtime $load_name"
                incr bt_curic
                if {$bt_curic >= $bt_totbc} {set bt_curic 0}
            }
        }
        Sync {
            set bt_next 0
        }
    }
    icserv_malf_cyclic
}
after $cyclicrate {cyclic_server}
}
```

```
# Procedure: getdata
proc getdata { varname} {
    global dataserv

    upvar $varname vname
    gets $dataserv vname
}
```

```
# Procedure: getdatavallist
proc getdatavallist { listid} {
    global DataSrcList
    global DataValList
    global dataserv

    putdata "gelist $listid"
    getdata DataValList
}
```

```
# Procedure: getval
proc getval { varname} {
    global dataserv
```

```
puts $dataserv "getval $varname"
flush $dataserv
gets $dataserv rtnval
return $rtnval
}
```

```

# Procedure: ics_get_diaglist
proc ics_get_diaglist { host } {
    global dispstat
    global argv0

    set namelist [array names dispstat]
    set rtnval ""
    set servname [file tail $argv0]
    set icsname "ics.tcl"
    set xfname "xfmain.tcl"
    foreach item $namelist {
        set spllilst [split $item :]
        set iname [lindex $spllilst 1]
        set hname [lindex $spllilst 0]
        if {$hname == $host} {
            if {($iname != $xfname) && ($iname != $servname) && ($iname != $icsname)} {lappend rtnval $iname}
        }
    }
    return $rtnval
}

```

```

# Procedure: ics_get_servers
proc ics_get_servers {} {
    global dispstat
    global argv0

    set namelist [array names dispstat]
    set rtnval ""
    set servname [file tail $argv0]
    foreach item $namelist {
        set spllilst [split $item :]
        set iname [lindex $spllilst 1]
        if {$iname == $servname} {lappend rtnval [lindex $spllilst 0]}
    }
    return $rtnval
}

```

```

# Procedure: icserv_dataLink
proc icserv_dataLink { loadname } {
    global dataserv
    global server_name

    if {[info exists dataserv]} { set dataserv "" }

    if {$dataserv == ""} {
        set dataserv [open "$server_name $loadname" r+]
        if {$dataserv == -1} {
            puts "Unable to connect $server_name to $loadname"
        }
    }
}

```

```

# Procedure: icserv_malf_cyclic
proc icserv_malf_cyclic {} {
    # This procedure assumes a calling rate of once per second
    #
    global mfbname
    global mfstate
    global mfdelta
    global mfinsev
    global mfcusev
    global mfdelay
    global mfrigger
    global trigval
    global statstr
    global malfstat

    if {$statstr == "Sync"} {
        # perform reset functions
        foreach name [array names mfstate] {

```

```

        unset mfbname($name)
        unset mfstate($name)
        unset mfdelta($name)
        unset mffinsev($name)
        unset mfcusev($name)
        unset mfdelay($name)
        unset mfrigger($name)
    }
    set malfst "mal: 0:0"
}
if {$statstr != "Run"} {return}

set malftotal 0
set malfactive 0
foreach name [array names mfstate] {
    incr malftotal
    if {$mfdelay($name) > 0} {
        incr mfdelay($name) -1
    }{
        if {$mfrigger($name) == ""} {
            if {$mfcusev($name) < $mffinsev($name)} {
                incr mfcusev($name) $mfdelta($name)
                if {$mfcusev($name) > $mffinsev($name)} {
                    set mfcusev($name) $mffinsev($name)
                }
                putdata "set $mfbname($name) $mfcusev($name)"
            }
            if {$mfstate($name)} {
                if {$name != $mfbname($name)} {
                    putdata "set $mfbname($name) $mfcusev($name)"
                }
                putdata "set $name 1"
                set mfstate($name) 0
            }
        } {if {$strigval($mfrigger($name))} {set mfrigger($name) ""}}
    }
    if {$mfstate($name) == 0} {incr malfactive}
}
set malfst "mal: ${malfactive};$malftotal"
}

```

```

# Procedure: icserv_titlebar_update
proc icserv_titlebar_update {} {
    global port_in
    global argv0

    # display connection information in window title
    set loadpath [lindex [service status] 5]
    wm title . "file tail $argv0 on port $port_in connected to: $loadpath"
}

```

```

# Procedure: incoming
proc incoming { socket machine port} {
    global socklist
    global enabled

    fconfigure $socket -blocking 1 -buffering full
    set name [lindex [fconfigure $socket -peername] 1]
    set enabled($name) 1
    .frame.text2 insert end "Connection requested from $name\n"
    lappend socklist $socket
    fileevent $socket readable "readmsg $socket"
}

```

```

# Procedure: isnatnum
proc isnatnum { value} {
    # This procedure checks whether value is a natural number
    # (i.e. an integer >= zero)
    #
    # Returns: 1 if a natural number
    #          0 if not a natural number
    #
}

```

```

set num $value
if {[catch {incr num}] {return 0}} {return 0}
if {$value < 0} {return 0}
return 1
}

```

```

# Procedure: kill
proc kill { fname } {
global dispstat
global feedback_cmd

if {[info exists dispstat($fname)]} {
# place quit command in feedback
set feedback_cmd([lindex $dispstat($fname) 0]) quit
unset dispstat($fname)
}
}

```

```

# Procedure: launch
proc launch { fname host } {
global dispstat
global feedback_cmd

if {[info exists dispstat($host:$fname)]} {
set feedback_cmd([lindex $dispstat($host:$fname) 0]) popup
}{
set haderror [catch {exec wish -file Data/$fname $host &} result]
if {$haderror != 0} {
puts $result
bell
}
}
}

```

```

# Procedure: putdata
proc putdata { line } {
global dataserv

puts $dataserv $line
flush $dataserv
}

```

```

# Procedure: quit
proc quit {} {
global connection
global list_id
global list_id2
global cyclic_on
global argv0

putdata exit
set cyclic_on 0

# close connections to other servers
foreach item [array names connection] {
if {$connection($item) != ""} {
css_cmd stat "newlist $list_id($item)"
css_cmd stat "newlist $list_id2($item)"
css_cmd stat "disconnect [info hostname]:[file tail $argv0]"
close $connection($item)
}
}
exit
}

```

```

# Procedure: readmsg
proc readmsg { socket } {
global DataSrcList
global DataValList
global history

```



```

global result
global errorctr
global enabled

set message ""
set totbytes 0
set line ""
set result ""

if {[eof $socket]} {
    close $socket
    return
}

# get initial information
set msgtype ""
set msg ""
catch {gets $socket msg}

if {$msg != ""} {
    set msgtype [lindex $msg 0]
    set line [lrange $msg 1 end]
    .frame.text2 insert end "$line\n"
}

set src [lindex [configure $socket -peername] 1]
switch $msgtype {
    read {
        if {$enabled($src)} {
            lappend history $line
            if {[catch {service $line} result]} {set result ""}
        }
    }
    ctrl {
        if {$enabled($src)} {
            if {[catch {service $line} result]} {set result ""}
        }
    }
    stat {
        if {[catch {service $line} result]} {set result ""}
    }
}

if {[eof $socket]} {
    close $socket
    return
} {
    # send copy of result string back to requester
    puts $socket $result
    catch {flush $socket}
}
.frame.text2 see end
}

```

```

# Procedure: service
proc service { cmdstr } {
    global load_host
    global load_name
    global DataSrcList
    global DataVallList
    global dispstat
    global feedback_cmd
    global dataserv
    global statusinfo
    global bt_interval
    global bt_next
    global bt_info
    global enabled

```

```

set rtrval ""
set cmd [lindex $cmdstr 0]
set arg1 [lindex $cmdstr 1]
set arg2 [lindex $cmdstr 2]
set arg3 [lindex $cmdstr 3]
set arg4 [lindex $cmdstr 4]
set arg5 [lindex $cmdstr 5]

```

```

set arg6 [lindex $cmdstr 6]
set arg7 [lindex $cmdstr 7]
set arg8 [lindex $cmdstr 8]
switch $cmd {
    bt_info {
        set bllist [lsort -integer [array names bt_info]]
        foreach item $bllist {lappend rtnval $bt_info($item)}
    }
    bt_int_get { set rtnval $bt_interval $arg1 }
    bt_int_set {
        if {[isnatnum $arg1]} {
            if {$arg1 < 14401} {
                set bt_interval $arg1
                set bt_next [getval seconds]
                incr bt_next $bt_interval
            }
        }
    }
    bt_reset {
        putdata "reset $arg1"
        set btnum [string range $arg1 2 3]
        set timestr [lindex $bt_info($btnum) 3]
        set totsecs [expr [clock scan $timestr] - [clock scan "00:00:00"]]
        set freq [getval freq]
        set totframes [expr ($totsecs * $freq) + 1]
        after 100
        putdata "set seconds $totsecs"
        putdata "set totframes $totframes"
    }
    disconnect {
        set clientname $arg1
        if {[info exists dispstat($clientname)]} {
            if {[string index $dispstat($clientname) 0] == "+"} {
                # Conducting a diagram replacement operation.
                # So don't delete the variable. Just remove the "+"
                set dispstat($clientname) [string range $dispstat($clientname) 1 end]
            } { unset dispstat($clientname) }
        }
        set cmachine [lindex [split $arg1 :] 0]
        if {[info exists enabled($cmachine)]} {
            unset enabled($cmachine)
        }
    }
    fast {
        putdata "fast $arg1"
    }
    findin {
        putdata "findin $arg1"
        getdata item
        while {$item != "#@eofind@"} {
            lappend rtnval $item
            getdata item
        }
    }
    findstart {
        putdata "findstart $arg1"
        getdata item
        while {$item != "#@eofind@"} {
            lappend rtnval $item
            getdata item
        }
    }
    frz {
        putdata freeze
    }
    getdiag { set rtnval [ics_get_diaglist $arg1] }
    getdisp {
        set dlist [lsort [array names dispstat]]
        foreach item $dlist {
            set line "$item $dispstat($item)"
            lappend rtnval $line
        }
    }
    getlist {
        set listno $arg1
    }
}

```

```

        getdatavallist $listno
        putdata "getlist $listno"
        getdata rtnval
        if      {[info exists feedback_cmd($listno)]} {
            lappend rtnval $feedback_cmd($listno)
            unset feedback_cmd($listno)
        }
    }
}

getpos {
    putdata "getpos $arg1"
    getdata rtnval
}

getservers { set rtnval [ics_get_servers] }

getval {
    set rtnval [getval $arg1]
}

kill {
    kill $arg1
}

launch {
    launch $arg1 $arg2
}

load {
    set name $load_name
    if {$name != ""} {
        exec $name &
        set load_name $name
        after 2000 {icserv_titlebar_update}
    }
}

mfinfo { set rtnval [service_mfinfo $arg1] }
mfmod { service_mfmod $arg1 $arg2 $arg3 $arg4 $arg5 $arg6 $arg7 $arg8 }
mfstat { set rtnval [service_mfstat $arg1] }

newlist {
    putdata newlist
    getdata rtnval
}

ptid_info {
    set rtnval [service_ptid_info $arg1 $arg2]
}

register {
    set clientname $arg1
    if      { $clientname != "" } {
        set listno $arg2
        set rchar ""
        if {[info exists dispstat($clientname)]} && [string first ics.tcl $clientname] == -1 {
            # Ensure only one copy of any item.
            # Delete old one.
            set orglist [lindex $dispstat($clientname) 0]
            set feedback_cmd($orglist) quit

            # Set a flag to indicate a diagram replacement is occurring
            set rchar "+*"
        }
        set dispstat($clientname) "${rchar}$listno [clock format [clock seconds] -format "%D %T"]"
    }
}

reset {
    putdata "reset $arg1 $arg2"
}

restrict { set enabled($arg1) $arg2 }
rinfo { set rtnval [service_rinfo $arg1] }
run {
    putdata run
}

setlist {
    set list_id [lindex $cmdstr 1]
    putdata "newlist $list_id"
    getdata junk
    set size [length $cmdstr]
    for {set ctr 2} {$ctr < $size} {incr ctr} {
        putdata "addtolist $list_id [lindex $cmdstr $ctr]"
    }
}

setpos {

```

```

        putdata "setpos $arg1 $arg2"
    }
    setval {
        putdata "set $arg1 $arg2"
    }
    slow {
        putdata "slow [[index $cmdstr 1]"
    }
    snap {
        putdata "snap [[index $cmdstr 1]"
    }
    status {
        set rtval $statusinfo
    }
    unload {
        putdata term
    }
}
return $rtval
}

```

```

# Procedure: service_mfinfo
proc service_mfinfo { mfname } {
    set rtval ""
    putdata "mfinfo $mfname"
    putdata "echo @eoinfo@"
    getdata line
    while {$line != "@eoinfo@"} {
        lappend rtval $line
        getdata line
    }
    return $rtval
}

```

```

# Procedure: service_mfmod
proc service_mfmod { name ffname state initsev ramptime finsev delay trigger } {
    global mffname
    global mfstate
    global mfdelta
    global mffinsev
    global mfcusev
    global mfdelay
    global mfrigger
    global mfdesc

```

```

if {$state == "OFF"} {
    if {[info exists mfstate($name)]} {
        putdata "set $name 0"
        unset mffname($name)
        unset mfstate($name)
        unset mfdelta($name)
        unset mffinsev($name)
        unset mfcusev($name)
        unset mfdelay($name)
        unset mfrigger($name)
        unset mfdesc($name)
    }
}

set mffname($name) $ffname
set mfstate($name) 1
if {[catch {expr $ramptime + 0} rtime]} {set rtime 0}
if {$rtime > 0} {
    if {[catch {expr ($finsev - $initsev) / $rtime} mfdelta($name)]} {set mfdelta($name) 0}
} {
    set mfdelta($name) 0
}
set mffinsev($name) $finsev
set mfcusev($name) $initsev
set mfdelay($name) $delay
set mfrigger($name) $trigger
putdata "mfinfo $name"
getdata line
set mfdesc($name) [[index $line end]]
}
}

```

```

# Procedure: service_mfstat
proc service_mfstat { name } {
  global mfbname
  global mfstate
  global mfdelta
  global mffinsev
  global mfcoursev
  global mfdelay
  global mfrigger
  global mfdesc

  if { $name == "all" } {
    set mlist [array names mfstate]
  } {
    if {[info exists mfstate($name)]} {
      set mlist $name
    } { set mlist "" }
  }
  set rtnval ""
  foreach item $mlist {
    set status "Active"
    set mfno [format "%-16s" $item]
    if { $mfdelay($item) > 0 } {
      set status [format "%-22s" "$mfdelay($item) second delay "]
    } {
      if { $mfrigger($item) != "" } {
        set status [format "%-22s" "waiting on trigger $mfrigger($item)"]
      }
    }
    lappend rtnval "$mfno $status $mfdesc($item)"
  }
  return $rtnval
}

```

```

# Procedure: service_ptid_info
proc service_ptid_info { vname info_type } {
  putdata "getinfo $vname"
  getdata ptid
  getdata briefdesc
  getdata units
  getdata system
  getdata type
  getdata dimensions
  getdata totbytes
  getdata partition
  getdata address
  getdata format
  getdata fulldesc
  putdata "echo @#endofinfo#@#"
  getdata line
  while { $line != "@#endofinfo#@#" } {
    append fulldesc $line
    getdata line
  }

  switch $info_type {
    ptid { set rtnval [index $ptid 2] }
    briefdesc { set rtnval [range $briefdesc 3 end] }
    units { set rtnval [index $units 2] }
    system { set rtnval [index $system 2] }
    type { set rtnval [index $type 2] }
    dimensions { set rtnval [index $dimensions 2] }
    totbytes { set rtnval [index $totbytes 3] }
    partition { set rtnval [index $partition 2] }
    address { set rtnval [index $address 2] }
    format { set rtnval [index $format 3] }
    fulldesc { set rtnval [range $fulldesc 3 end] }
    default { set rtnval "" }
  }
  return $rtnval
}

```



```

# Sideeffects: none
#####
global tcl_platform
if {$tcl_platform(platform) == "unix"} {exec cat $filename} {
    set fileid [open $filename "r"]
    set data [read $fileid]
    close $fileid
    return $data
}
}
}

# Procedure: Chmod
if {[info procs Chmod] == ""} {
proc Chmod { mode file} {
# xf ignore me 7
#####
# Procedure: Chmod
# Description: ignore UNIX chmod under DOS
# Arguments: file1 file2/directory
# Returns: nothing
# Sideeffects: none
#####
global tcl_platform
if {$tcl_platform(platform) == "unix"} {eval exec chmod $mode $file} {
    regsub -all {} $file1 {} file1
    regsub -all {} $file2 {} file2
    eval exec command.com /c copy /y $file1 $file2 >@stderr
}
}
}

# Procedure: Cp
if {[info procs Cp] == ""} {
proc Cp { file1 file2} {
# xf ignore me 7
#####
# Procedure: Cp
# Description: emulate UNIX cp with DOS COPY
# Arguments: file1 file2/directory
# Returns: nothing
# Sideeffects: none
#####
global tcl_platform
if {$tcl_platform(platform) == "unix"} {eval exec cp $file1 $file2} {
    regsub -all {} $file1 {} file1
    regsub -all {} $file2 {} file2
    eval exec command.com /c copy /y $file1 $file2 >@stderr
}
}
}

# Procedure: GetSelection
if {[info procs GetSelection] == ""} {
proc GetSelection {} {
# xf ignore me 7
#####
# Procedure: GetSelection
# Description: get current selection
# Arguments: none
# Returns: none
# Sideeffects: none
#####

# the save way
set xfSelection ""
catch "selection get" xfSelection
if {"$xfSelection" == "selection doesn't exist or form !\"STRING!\" not defined"} {
    return ""
} {
    return $xfSelection
}
}
}
}

```

```

}
}

# Procedure: Ls
if {[info procs Ls] == ""} {
proc Ls { args } {
# xf ignore me 7
#####
# Procedure: Ls
# Description: emulate UNIX ls
# Arguments: like UNIX (switches authorized -F -a {-a is ignored})
# Returns: directory list
# Sideeffects: none
#####
global tcl_platform
if {$tcl_platform(platform) == "unix"} {eval exec ls $args} {
set last [lindex $args end]
if {[search $last -* ] >= 0 || $last == ""} {set path ""} {set path $last/*}
set lst [glob $path]
set lst1 ""
if {[search -exact $args "-F"] >= 0} then {set car "/" } else {set car ""}
foreach f $lst {
if {[file isdirectory $f]} {set f [file tail $f]$car} {set f [file tail $f]}
lappend lst1 $f
}
return [lsort $lst1]
}
}
}
}

```

```

# Procedure: NoFunction
if {[info procs NoFunction] == ""} {
proc NoFunction { args } {
# xf ignore me 7
#####
# Procedure: NoFunction
# Description: do nothing (especially with scales and scrollbars)
# Arguments: args - a number of ignored parameters
# Returns: none
# Sideeffects: none
#####
}
}
}

```

```

# Procedure: Rm
if {[info procs Rm] == ""} {
proc Rm { filename } {
# xf ignore me 7
#####
# Procedure: Rm
# Description: emulate UNIX rm with DOS DEL
# Arguments: filename(s)
# Returns: nothing
# Sideeffects: none
#####
global tcl_platform
if {$tcl_platform(platform) == "unix"} {exec rm -f $filename} {
regsub -all {/} $filename {\\} filename
eval exec command.com /c del $filename >@stderr
}
}
}
}

```

```

# Procedure: SN
if {[info procs SN] == ""} {
proc SN { {xfName ""} } {
# xf ignore me 7
#####
# Procedure: SN
# Description: map a symbolic name to the widget path
# Arguments: xfName

```



```
# Returns: the symbolic name
# Sideeffects: none
#####
```

```
SymbolicName $xfName
}
}
```

```
# Procedure: SymbolicName
if {[info procs SymbolicName] == ""} {
proc SymbolicName { {xfName ""} } {
# xf ignore me 7
#####
# Procedure: SymbolicName
# Description: map a symbolic name to the widget path
# Arguments: xfName
# Returns: the symbolic name
# Sideeffects: none
#####
```

```
global symbolicName

if {"$xfName" != ""} {
set xfArrayName ""
append xfArrayName symbolicName ( $xfName )
if {[catch "set \"$xfArrayName\" xfValue"]} {
return $xfValue
} {
if {[info commands XFProcError] != ""} {
XFProcError "Unknown symbolic name:\n$xfName"
} {
puts stderr "XF error: unknown symbolic name:\n$xfName"
}
}
}
return ""
}
}
```

```
# Procedure: Unalias
if {[info procs Unalias] == ""} {
proc Unalias { aliasName } {
# xf ignore me 7
#####
# Procedure: Unalias
# Description: remove an alias for a procedure
# Arguments: aliasName - the alias name to remove
# Returns: none
# Sideeffects: internalAliasList is updated, and the alias
# proc is removed
#####
global internalAliasList
```

```
set xfIndex [lsearch $internalAliasList "$aliasName ""]
if {$xfIndex != -1} {
rename $aliasName ""
set internalAliasList [lreplace $internalAliasList $xfIndex $xfIndex]
}
}
}
```

```
# application parsing procedure
proc XFLocalParseAppDeFs {xfAppDefFile} {
global xfAppDefaults tcl_platform
```

```
# basically from: Michael Moore
if {[file exists $xfAppDefFile] &&
[file readable $xfAppDefFile] &&
[file type $xfAppDefFile] == "link"} {
catch "file type $xfAppDefFile" xfType
while {"$xfType" == "link"} {
```

```

if {[catch "file readlink $xfAppDefFile" xfAppDefFile]} {
    return
}
catch "file type $xfAppDefFile" xfType
}
}
if {"$xfAppDefFile" != "" &&
    [file exists $xfAppDefFile] &&
    [file readable $xfAppDefFile] &&
    "[file type $xfAppDefFile]" == "file"} {
    return
}
if {[catch "open $xfAppDefFile r" xfResult]} {
    set xfAppFileContents [read $xfResult]
    close $xfResult
    foreach line [split $xfAppFileContents "\n"] {
        # backup indicates how far to backup. It applies to the
        # situation where a resource name ends in . and when it
        # ends in *. In the second case you want to keep the *
        # in the widget name for pattern matching, but you want
        # to get rid of the . if it is the end of the name.
        set backup -2
        set line [string trim $line]
        if {[string index $line 0] == "#"} {
            # skip comments and empty lines
            continue
        }
        if {[string compare "windows" $tcl_platform(platform)]} {
            set list [split $line ";"]
        } {
            set list [split $line "."]
        }
        set resource [string trim [lindex $list 0]]
        set i [string last "." $resource]
        set j [string last "*" $resource]
        if {$j > $i} {
            set i $j
            set backup -1
        }
        incr i
        set name [string range $resource $i end]
        incr i $backup
        set widname [string range $resource 0 $i]
        set value [string trim [lindex $list 1]]
        if {"$widname" != "" && "$widname" != ""} {
            # insert the widget and resourcename to the application
            # defaults list.
            if {[info exists xfAppDefaults]} {
                set xfAppDefaults ""
            }
            lappend xfAppDefaults [list $widname [string tolower $name] $value]
        }
    }
}
}
}

# application loading procedure
proc XFLocalLoadAppDefs {xfClasses "" {xfPriority "startupFile"} {xfAppDefFile ""} {
    global env tcl_platform

    if {[string compare "windows" $tcl_platform(platform)]} {
        set separator ";"
    } {
        set separator "."
    }
    if {"$xfAppDefFile" == ""} {
        set xfFileList ""
        if {[info exists env(XUSERFILESEARCHPATH)]} {
            eval lappend xfFileList [split $env(XUSERFILESEARCHPATH) $separator]
        }
        if {[info exists env(XAPPLRESDIR)]} {
            eval lappend xfFileList [split $env(XAPPLRESDIR) $separator]
        }
        if {[info exists env(XFILESEARCHPATH)]} {
            eval lappend xfFileList [split $env(XFILESEARCHPATH) $separator]
        }
    }
}

```



```

        close $fin
    }

set odspath [get_odspath]

# set up load name and TCP/IP socket port number
set load_name [lindex $argv 0]
if {$load_name == ""} {set load_name rtsched.exe}
set port_in [lindex $argv 1]
if {$port_in == ""} {set port_in 5000}

## check for valid port number
set temp $port_in
if {[catch {incr temp} result]} {set port_in 5000}
if {$port_in < 5000 || $port_in > 5999} {set port_in 5000}

.frame.text2 delete 1.0 end
.frame.text2 insert 1.0 "Starting message server\n"

if {[info exists enabled]} {unset enabled}
set socklist ""
set sockflag [socket -server { incoming } $port_in]

# clear command history
set history ""

#clear feedback commands
foreach item [array names feedback_cmd] {
    unset feedback_cmd($item)
}

#clear out display tracking information
if {[info exists dispstat]} {unset dispstat}

# connect to data server program
if {[info exists DataSrcList]} {unset DataSrcList}
set dataserv ""
icserv_dataLink $load_name

# register this server
if {$load_host == $hostname} {
    set dispstat(${hostname}:file tail $argv0)) "0 [clock format [clock seconds] -format "%D %T"]"
    {
        set DataSrcList ""
        set DataSrcList2 ""
        switchdest $load_host
    }
}

# set up for backtrack
set bt_interval 60
set bt_next 0

# display connection information in window title
icserv_titlebar_update

set cyclic_on 1
cyclic_server
}

# prepare auto loading
global auto_path
global tk_library
global xflLoadPath
foreach xfElement [eval list [split $xfLoadPath @] $auto_path] {
    if {[file exists $xfElement/tclIndex]} {
        lappend auto_path $xfElement
    }
}

# initialize global variables
proc InitGlobals {} {
    global {}
    set {} {1}
    global {DataValList}

```

set {DataValList} { ----- 0.0059509 -----

```
global {bt_curic}
set {bt_curic} {37}
global {bt_info}
set {bt_info(0)} { 00 10/03/97 21:05:12 02:04:55 rtsched.exe}
set {bt_info(1)} { 01 10/03/97 21:06:11 02:05:55 rtsched.exe}
set {bt_info(10)} { 10 10/03/97 21:15:10 02:14:55 rtsched.exe}
set {bt_info(11)} { 11 10/03/97 21:16:09 02:15:55 rtsched.exe}
set {bt_info(12)} { 12 10/03/97 21:17:09 02:16:55 rtsched.exe}
set {bt_info(13)} { 13 10/03/97 21:18:08 02:17:55 rtsched.exe}
set {bt_info(14)} { 14 10/03/97 21:19:08 02:18:55 rtsched.exe}
set {bt_info(15)} { 15 10/03/97 21:20:08 02:19:55 rtsched.exe}
set {bt_info(16)} { 16 10/03/97 21:21:08 02:20:55 rtsched.exe}
set {bt_info(17)} { 17 10/03/97 21:22:08 02:21:55 rtsched.exe}
set {bt_info(18)} { 18 10/03/97 21:23:07 02:22:55 rtsched.exe}
set {bt_info(19)} { 19 10/03/97 21:24:07 02:23:55 rtsched.exe}
set {bt_info(2)} { 02 10/03/97 21:07:11 02:06:55 rtsched.exe}
set {bt_info(20)} { 20 10/03/97 21:25:07 02:24:55 rtsched.exe}
set {bt_info(21)} { 21 10/03/97 21:26:07 02:25:55 rtsched.exe}
set {bt_info(22)} { 22 10/03/97 21:27:06 02:26:55 rtsched.exe}
set {bt_info(23)} { 23 10/03/97 21:28:06 02:27:55 rtsched.exe}
set {bt_info(24)} { 24 10/03/97 21:29:06 02:28:55 rtsched.exe}
set {bt_info(25)} { 25 10/03/97 21:30:05 02:29:55 rtsched.exe}
set {bt_info(26)} { 26 10/03/97 21:31:05 02:30:55 rtsched.exe}
set {bt_info(27)} { 27 10/03/97 21:32:05 02:31:55 rtsched.exe}
set {bt_info(28)} { 28 10/03/97 21:33:05 02:32:55 rtsched.exe}
set {bt_info(29)} { 29 10/03/97 21:34:04 02:33:55 rtsched.exe}
set {bt_info(3)} { 03 10/03/97 21:08:11 02:07:55 rtsched.exe}
set {bt_info(30)} { 30 10/03/97 21:35:04 02:34:55 rtsched.exe}
set {bt_info(31)} { 31 10/03/97 21:36:04 02:35:55 rtsched.exe}
set {bt_info(32)} { 32 10/03/97 21:37:04 02:36:55 rtsched.exe}
set {bt_info(33)} { 33 10/03/97 21:38:04 02:37:55 rtsched.exe}
set {bt_info(34)} { 34 10/03/97 21:39:03 02:38:55 rtsched.exe}
set {bt_info(35)} { 35 10/03/97 21:40:03 02:39:55 rtsched.exe}
set {bt_info(36)} { 36 10/03/97 21:41:03 02:40:55 rtsched.exe}
set {bt_info(37)} { 37 10/03/97 20:42:03 01:41:41 rtsched.exe}
set {bt_info(38)} { 38 10/03/97 20:43:03 01:42:41 rtsched.exe}
set {bt_info(39)} { 39 10/03/97 20:44:03 01:43:41 rtsched.exe}
set {bt_info(4)} { 04 10/03/97 21:09:11 02:08:55 rtsched.exe}
set {bt_info(40)} { 40 10/03/97 20:45:02 01:44:41 rtsched.exe}
set {bt_info(41)} { 41 10/03/97 20:46:15 01:45:54 rtsched.exe}
set {bt_info(42)} { 42 10/03/97 20:47:15 01:46:54 rtsched.exe}
set {bt_info(43)} { 43 10/03/97 20:48:15 01:47:54 rtsched.exe}
set {bt_info(44)} { 44 10/03/97 20:49:14 01:48:54 rtsched.exe}
set {bt_info(45)} { 45 10/03/97 20:50:14 01:49:54 rtsched.exe}
set {bt_info(46)} { 46 10/03/97 20:51:14 01:50:54 rtsched.exe}
set {bt_info(47)} { 47 10/03/97 20:52:14 01:51:54 rtsched.exe}
set {bt_info(48)} { 48 10/03/97 20:53:13 01:52:54 rtsched.exe}
set {bt_info(49)} { 49 10/03/97 20:54:14 01:53:54 rtsched.exe}
set {bt_info(5)} { 05 10/03/97 21:10:10 02:09:55 rtsched.exe}
set {bt_info(50)} { 50 10/03/97 20:55:13 01:54:54 rtsched.exe}
set {bt_info(51)} { 51 10/03/97 20:56:14 01:55:55 rtsched.exe}
set {bt_info(52)} { 52 10/03/97 20:57:13 01:56:55 rtsched.exe}
set {bt_info(53)} { 53 10/03/97 20:58:13 01:57:55 rtsched.exe}
set {bt_info(54)} { 54 10/03/97 20:59:13 01:58:55 rtsched.exe}
set {bt_info(55)} { 55 10/03/97 21:00:13 01:59:55 rtsched.exe}
set {bt_info(56)} { 56 10/03/97 21:01:12 02:00:55 rtsched.exe}
set {bt_info(57)} { 57 10/03/97 21:02:12 02:01:55 rtsched.exe}
set {bt_info(58)} { 58 10/03/97 21:03:12 02:02:55 rtsched.exe}
set {bt_info(59)} { 59 10/03/97 21:04:11 02:03:55 rtsched.exe}
set {bt_info(6)} { 06 10/03/97 21:11:10 02:10:55 rtsched.exe}
set {bt_info(7)} { 07 10/03/97 21:12:10 02:11:55 rtsched.exe}
set {bt_info(8)} { 08 10/03/97 21:13:09 02:12:55 rtsched.exe}
set {bt_info(9)} { 09 10/03/97 21:14:09 02:13:55 rtsched.exe}
global {bt_interval}
set {bt_interval} {60}
global {bt_next}
set {bt_next} {0}
global {bt_totbc}
set {bt_totbc} {60}
global {cyclic_on}
set {cyclic_on} {1}
global {cyclicrate}
set {cyclicrate} {1000}
```

```
global {dataserv}
set {dataserv} {file32}
global {dispslat}
set {dispslat(asus:xfmain.tcl)} {2 10/03/97 21:42:52}
global {enabled}
global {errorctr}
set {errorctr} {0}
global {feedback_cmd}
set {feedback_cmd(0)} {quit}
global {history}
set {history} {unload load {reset IC50}}
global {hostname}
set {hostname} {asus}
global {junk}
set {junk} {1}
global {load_host}
set {load_host} {asus}
global {load_name}
set {load_name} {rtsched.exe}
global {machine}
set {machine} {cyrix}
global {malfstal}
set {malfstal} {malf: 0:0}
global {odspath}
set {odspath} {s:/ods1st}
global {port}
set {port} {5002}
global {port_in}
set {port_in} {5000}
global {restricted}
set {restricted(asus)} {1}
global {result}
set {result} {}
global {server_name}
set {server_name} {rldata.exe}
global {simtime}
set {simtime} {00:00:00}
global {sockflag}
set {sockflag} {sock92}
global {socklist}
set {socklist} {sock176}
global {statstr}
set {statstr} {Freeze}
global {statusinfo}
set {statusinfo} {Freeze 1.00 00:00:00 0 0.000000 {s:\demo\rtsched.exe} IC50 60 {malf: 0:0}}
global {trigval}
set {trigval(1)} {1}
set {trigval(10)} {1}
set {trigval(2)} {1}
set {trigval(3)} {1}
set {trigval(4)} {1}
set {trigval(5)} {1}
set {trigval(6)} {1}
set {trigval(7)} {1}
set {trigval(8)} {1}
set {trigval(9)} {1}
```

please don't modify the following
variables. They are needed by xf.

```
global {autoLoadList}
set {autoLoadList(icserver.tcl)} {0}
set {autoLoadList(main.tcl)} {0}
global {internalAliasList}
set {internalAliasList} {}
global {moduleList}
set {moduleList(icserver.tcl)} {}
global {preloadList}
set {preloadList(xfInternal)} {}
global {symbolicName}
set {symbolicName(root)} {,}
global {xfWmSetPosition}
set {xfWmSetPosition} {}
global {xfWmSetSize}
set {xfWmSetSize} {}
global {xfAppDefToplevels}
```

```

set {xfAppDefToplevels} {}
}

# initialize global variables
InitGlobals

# display/remove toplevel windows.
ShowWindow.

# load default bindings.
if {[info exists env(XF_BIND_FILE)]} &&
    "[info procs XFShowHelp]" == "" {
    source $env(XF_BIND_FILE)
}

# initialize bindings for all widgets.
XFInitAllBindings

# parse and apply application defaults.
XFLocalLoadAppDefs IcsServer
XFLocalSetAppDefs

# end source
EndSrc

# eof
#

#!wish -f
# Program: icsgraph
# Tcl version: 7.6 (Tcl/Tk/XF)
# Tk version: 4.2
# XF version: 4.0
#
# -----
# Copyright (C) 1996 Dennis R. LaBelle (drlabel@albany.net) All Rights Reserved.
#
# PERMISSION IS GRANTED TO DISTRIBUTE THIS SOFTWARE FREELY, WITH THE
# EXCEPTION THAT ONE MAY NOT CHARGE FOR IT OR INCLUDE IT WITH SOFTWARE
# WHICH IS SOLD. Permission to use, copy, modify, and distribute this software and
# its documentation for educational, research, internal corporate and non-profit
# purposes, without fee, and without a written agreement is hereby granted for all
# cases that do not conflict with the restriction in the first sentence of this
# paragraph, provided that the above copyright notice, this paragraph, and the
# following three paragraphs appear in all copies.
#
# Permission to incorporate this software into commercial products may be obtained
# from the author (e-mail drlabel@albany.net).
#
# IN NO EVENT SHALL THE AUTHOR BE LIABLE TO ANY PARTY FOR DIRECT,
# INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF
# THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE AUTHOR HAS
# BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
# THE AUTHOR SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
# PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS"
# BASIS, AND THE AUTHOR HAS NO OBLIGATION TO PROVIDE MAINTENANCE,
# SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
# -----

# module inclusion
global env
global xflLoadPath
global xflLoadInfo
set xflMisc(separator) @
set xflLoadInfo 0
if {[info exists env(XF_LOAD_PATH)]} {
    if {[string first $env(XF_LOAD_PATH) @$flib] == -1} {
        set xflLoadPath $env(XF_LOAD_PATH):@$flib
    } {
        set xflLoadPath @$flib
    }
}

```



```

}{
set xflLoadPath ./lib
}

global argc
global argv
set tmpArgv ""
for {set counter 0} {$counter < $argc} {incr counter 1} {
case [string tolower [lindex $argv $counter]] in {
{-xflloadpath} {
incr counter 1
set xflLoadPath "[lindex $argv $counter]:$xflLoadPath"
}
{-xflstartup} {
incr counter 1
source [lindex $argv $counter]
}
{-xflbindfile} {
incr counter 1
set env(XF_BIND_FILE) "[lindex $argv $counter]"
}
{-xflcolorfile} {
incr counter 1
set env(XF_COLOR_FILE) "[lindex $argv $counter]"
}
{-xflcursorfile} {
incr counter 1
set env(XF_CURSOR_FILE) "[lindex $argv $counter]"
}
{-xflfontfile} {
incr counter 1
set env(XF_FONT_FILE) "[lindex $argv $counter]"
}
{-xflmodelmono} {
tk colormodel . monochrome
}
{-xflmodelcolor} {
tk colormodel . color
}
{-xflloading} {
set xflLoadInfo 1
}
{-xflnloading} {
set xflLoadInfo 0
}
{default} {
lappend tmpArgv [lindex $argv $counter]
}
}
}
set argv $tmpArgv
set argc [length $tmpArgv]
unset counter
unset tmpArgv

```

```

# procedure to show window ShowWindow.query
proc ShowWindow.query { args } {
# xf ignore me 7

```

```

# build widget .query
if {"[info procs XFedit]" != ""} {
catch "XFDestroy .query"
}
catch "destroy .query"
}
toplevel .query

```

```

# Window manager configurations

```

```

wm positionfrom .query program
wm sizefrom .query program
wm maxsize .query 1000 864
wm minsize .query 374 300
wm protocol .query WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}

```

```

wm title .query {Select parameter for Graph 1}

# build widget .query.frame
frame .query.frame

# build widget .query.frame.scrollbar3
scrollbar .query.frame.scrollbar3 -command {.query.frame.listbox1 xview} -orient {horizontal} -relief {raised}

# build widget .query.frame.scrollbar2
scrollbar .query.frame.scrollbar2 -command {.query.frame.listbox1 yview} -relief {raised}

# build widget .query.frame.listbox1
listbox .query.frame.listbox1 -background {lightblue} -font {-adobe-courier-bold-r-normal--14-140-75-m-90-iso8859-1} -relief {raised} -xscrollcommand
{.query.frame.scrollbar3 set} -yscrollcommand {.query.frame.scrollbar2 set}
# bindings
bind .query.frame.listbox1 <Double-Button-1> {set index [%W nearest %y]}
if {$index != ""} {
    set param($curparm) ([index [%W get $index ] 0])
    set gnum ([index {wm title .query} end])
    incr gnum -1
    sendparm $gnum $curparm
}
}

# build widget .query.frame3
frame .query.frame3 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .query.frame3.button4
button .query.frame3.button4 -background {slategrey} -command {DestroyWindow.query} -font {-adobe-helvetica-bold-r-normal--14-140-75-p-82-iso8859-1}
-foreground {yellow} -padx {11} -pady {4} -text {Done}

# build widget .query.frame10
frame .query.frame10 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .query.frame10.frame2
frame .query.frame10.frame2 -height {30} -width {30}

# build widget .query.frame10.frame2.label5
label .query.frame10.frame2.label5 -background {steelblue} -font {-adobe-helvetica-bold-o-normal--12-120-75-p-69-iso8859-1} -foreground {white} -relief {raised} -text
{ Match; }

# build widget .query.frame10.frame2.entry6
entry .query.frame10.frame2.entry6 -background {lightblue} -font {-adobe-helvetica-bold-o-normal--12-120-75-p-69-iso8859-1} -textvariable {query_match}
# bindings
bind .query.frame10.frame2.entry6 <Key-Return> {query_list $query_match}

# build widget .query.frame10.frame11
frame .query.frame10.frame11 -height {30} -width {30}

# build widget .query.frame10.frame11.radiobutton12
radiobutton .query.frame10.frame11.radiobutton12 -font {-adobe-helvetica-bold-r-normal--12-120-75-p-70-iso8859-1} -text {items starting with} -value {start} -variable
{query_type}

# build widget .query.frame10.frame11.radiobutton13
radiobutton .query.frame10.frame11.radiobutton13 -font {-adobe-helvetica-bold-r-normal--12-120-75-p-70-iso8859-1} -text {items containing} -value {in} -variable
{query_type}

# pack master .query.frame
pack configure .query.frame.scrollbar2 -fill y -side right
pack configure .query.frame.listbox1 -expand 1 -fill both
pack configure .query.frame.scrollbar3 -fill x -side bottom

# pack master .query.frame3
pack configure .query.frame3.button4 -expand 1 -fill x -side left

# pack master .query.frame10
pack configure .query.frame10.frame2 -fill x
pack configure .query.frame10.frame11 -fill x

# pack master .query.frame10.frame2
pack configure .query.frame10.frame2.label5 -side left
pack configure .query.frame10.frame2.entry6 -expand 1 -fill x

# pack master .query.frame10.frame11
pack configure .query.frame10.frame11.radiobutton12 -expand 1 -fill x -side left

```

```
pack configure .query.frame10.frame11.radiobutton13 -expand 1 -fill x -side left
```

```
# pack master .query
pack configure .query.frame10 -fill x
pack configure .query.frame -expand 1 -fill both
pack configure .query.frame3 -fill x
```

```
.query.frame.listbox1 insert end {ed_htr1on      Heater 1 on}
.query.frame.listbox1 insert end {ed_htr2on      Heater 2 on}
.query.frame.listbox1 insert end {ed_htr3on      Heater 3 on}
.query.frame.listbox1 insert end {edibattery     Battery current}
.query.frame.listbox1 insert end {edivoltreg     DC volt regulator current}
.query.frame.listbox1 insert end {edkrhr1        Heater 1 resistance}
.query.frame.listbox1 insert end {edkrhr2        Heater 2 resistance}
.query.frame.listbox1 insert end {edkrhr3        Heater 3 resistance}
.query.frame.listbox1 insert end {edkrvoltreg    Voltage reg. internal resistance}
.query.frame.listbox1 insert end {edravgcell     Average battery cell resistance}
.query.frame.listbox1 insert end {edrbattery     Battery resistance}
.query.frame.listbox1 insert end {edrheaters_on  Total resistance from active hea}
.query.frame.listbox1 insert end {edtotbattcells Number of cells in the battery}
.query.frame.listbox1 insert end {edvavgcell     Average cell voltage}
.query.frame.listbox1 insert end {edvbattery     Battery interval voltage}
.query.frame.listbox1 insert end {edvbus1       Bus 1 voltage}
.query.frame.listbox1 insert end {edvreg        Voltage regulator setpoint}
```

```
if {[info procs XFEdit] != ""} {
  catch "XFMiscBindWidgetTree .query"
  after 2 "catch {XFEditSetShowWindows}"
}
}
```

```
proc DestroyWindow.query {} {# xf ignore me 7
  if {[info procs XFEdit] != ""} {
    if {[info commands .query] != ""} {
      global xfShowWindow.query
      set xfShowWindow.query 0
      XFEditSetPath .
      after 2 "XFSaveAsProc .query; XFEditSetShowWindows"
    }
  }
  catch "destroy .query"
  update
}
}
```

```
# procedure to show window ShowWindow.range
```

```
proc ShowWindow.range { args } {
  # xf ignore me 7
```

```
# build widget .range
if {[info procs XFEdit] != ""} {
  catch "XFDestroy .range"
}
catch "destroy .range"
}
toplevel .range -relief {raised}
```

```
# Window manager configurations
wm positionfrom .range program
wm sizefrom .range program
wm geometry .range 320x171+587+2
wm maxsize .range 320 171
wm minsize .range 320 171
wm protocol .range WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .range {Select display range for Graph 1}
```

```
# build widget .range.frame4
frame .range.frame4 -borderwidth {2} -height {30} -relief {ridge} -width {30}
```

```
# build widget .range.frame4.button6
```

```

button .range.frame4.button6 -background {slategrey} -command {set gnum [lindex [wm title .range] end]
incr gnum -1
set tailsize($gnum) $tailsizetmp
set dispmode($gnum) $disprange
set yscalemode($gnum) $yscale
selranges $gnum) -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1} -foreground {yellow} -padx {11} -pady {4} -text {Apply}

# build widget .range.frame4.button7
button .range.frame4.button7 -background {slategrey} -command {DestroyWindow.range} -font {-adobe-helvetica-medium-r-normal-14-140-75-75-p-77-iso8859-1}
-foreground {yellow} -padx {11} -pady {4} -state {active} -text {Done}

# build widget .range.frame6
frame .range.frame6 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .range.frame6.frame2
frame .range.frame6.frame2 -height {30} -width {30}

# build widget .range.frame6.frame2.entry2
entry .range.frame6.frame2.entry2 -background {lightblue} -textvariable {tailsizetmp} -width {7}

# build widget .range.frame6.frame2.radiobutton3
radiobutton .range.frame6.frame2.radiobutton3 -text {Tail end (secs) } -value {tail} -variable {disprange} -width {15}

# build widget .range.frame6.frame5
frame .range.frame6.frame5 -height {30} -width {30}

# build widget .range.frame6.frame5.radiobutton3
radiobutton .range.frame6.frame5.radiobutton3 -text {Data range (secs)} -value {range} -variable {disprange} -width {15}

# build widget .range.frame6.frame5.frame9
frame .range.frame6.frame5.frame9 -height {30} -width {30}

# build widget .range.frame6.frame5.frame9.label7
label .range.frame6.frame5.frame9.label7 -borderwidth {0} -foreground {blue} -text {From;}

# build widget .range.frame6.frame5.frame9.entry2
entry .range.frame6.frame5.frame9.entry2 -background {lightblue} -textvariable {rangestart} -width {7}

# build widget .range.frame6.frame5.frame9.label8
label .range.frame6.frame5.frame9.label8 -borderwidth {0} -foreground {blue} -text {To;}

# build widget .range.frame6.frame5.frame9.entry6
entry .range.frame6.frame5.frame9.entry6 -background {lightblue} -textvariable {rangestop} -width {7}

# build widget .range.frame6.radiobutton11
radiobutton .range.frame6.radiobutton11 -text {All data} -value {all} -variable {disprange}

# build widget .range.frame9
frame .range.frame9 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .range.frame9.radiobutton3
radiobutton .range.frame9.radiobutton3 -text {Auto value scale } -value {auto} -variable {yscale} -width {15}

# build widget .range.frame9.frame11
frame .range.frame9.frame11 -height {30} -width {30}

# build widget .range.frame9.frame11.radiobutton3
radiobutton .range.frame9.frame11.radiobutton3 -text {Value scale } -value {range} -variable {yscale} -width {15}

# build widget .range.frame9.frame11.frame9
frame .range.frame9.frame11.frame9 -height {30} -width {30}

# build widget .range.frame9.frame11.frame9.label7
label .range.frame9.frame11.frame9.label7 -borderwidth {0} -foreground {blue} -text {From;}

# build widget .range.frame9.frame11.frame9.entry2
entry .range.frame9.frame11.frame9.entry2 -background {lightblue} -textvariable {yscalemin} -width {7}

# build widget .range.frame9.frame11.frame9.label8
label .range.frame9.frame11.frame9.label8 -borderwidth {0} -foreground {blue} -text {To;}

# build widget .range.frame9.frame11.frame9.entry6
entry .range.frame9.frame11.frame9.entry6 -background {lightblue} -textvariable {yscalemax} -width {7}

# pack master .range.frame4

```

```
pack configure .range.frame4.button6 -expand 1 -fill both -side left
pack configure .range.frame4.button7 -expand 1 -fill both -side left
```

```
# pack master .range.frame6
pack configure .range.frame6.radiobutton11 -anchor w -ipadx 3
pack configure .range.frame6.frame2 -fill x
pack configure .range.frame6.frame5 -fill x
```

```
# pack master .range.frame6.frame2
pack configure .range.frame6.frame2.radiobutton3 -fill x -side left
pack configure .range.frame6.frame2.entry2 -side left
```

```
# pack master .range.frame6.frame5
pack configure .range.frame6.frame5.radiobutton3 -side left
pack configure .range.frame6.frame5.frame9 -expand 1 -fill x -side left
```

```
# pack master .range.frame6.frame5.frame9
pack configure .range.frame6.frame5.frame9.label7 -side left
pack configure .range.frame6.frame5.frame9.entry2 -expand 1 -fill x -side left
pack configure .range.frame6.frame5.frame9.label8 -side left
pack configure .range.frame6.frame5.frame9.entry6 -expand 1 -fill x -side left
```

```
# pack master .range.frame9
pack configure .range.frame9.radiobutton3 -anchor w
pack configure .range.frame9.frame11 -fill x
```

```
# pack master .range.frame9.frame11
pack configure .range.frame9.frame11.radiobutton3 -side left
pack configure .range.frame9.frame11.frame9 -fill x
```

```
# pack master .range.frame9.frame11.frame9
pack configure .range.frame9.frame11.frame9.label7 -side left
pack configure .range.frame9.frame11.frame9.entry2 -expand 1 -fill x -side left
pack configure .range.frame9.frame11.frame9.label8 -side left
pack configure .range.frame9.frame11.frame9.entry6 -expand 1 -fill x -side left
```

```
# pack master .range
pack configure .range.frame6 -fill x
pack configure .range.frame9 -fill x
pack configure .range.frame4 -fill both
```

```
if {"[info procs XFEdit]" != ""} {
  catch "XFMiscBindWidgetTree .range"
  after 2 "catch {XFEditSetShowWindows}"
}
}
```

```
proc DestroyWindow.range {} {# xf ignore me 7
if {"[info procs XFEdit]" != ""} {
  if {"[info commands .range]" != ""} {
    global xfShowWindow.range
    set xfShowWindow.range 0
    XFEditSetPath .
    after 2 "XFSaveAsProc .range; XFEditSetShowWindows"
  }
} {
  catch "destroy .range"
  update
}
}
```

```
# procedure to show window ShowWindow.samplerate
proc ShowWindow.samplerate { args } {
# xf ignore me 7
```

```
# build widget .samplerate
if {"[info procs XFEdit]" != ""} {
  catch "XFDestroy .samplerate"
} {
  catch "destroy .samplerate"
}
}
toplevel .samplerate -relief {raised}
```

```
# Window manager configurations
```

```

wm positionfrom .samplerate program
wm sizefrom .samplerate program
wm geometry .samplerate 302x112+124+239
wm maxsize .samplerate 302 112
wm minsize .samplerate 302 112
wm protocol .samplerate WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title .samplerate {Select Sampling Rate}

# build widget .samplerate.frame2
frame .samplerate.frame2 -height {30} -width {30}

# build widget .samplerate.frame2.entry2
entry .samplerate.frame2.entry2 -background {lightblue} -textvariable {interval}

# build widget .samplerate.frame2.radiobutton3
radiobutton .samplerate.frame2.radiobutton3 -text {User specified interval (msecs)} -value {userspec} -variable {freq_new}

# build widget .samplerate.frame3
frame .samplerate.frame3 -height {30} -width {30}

# build widget .samplerate.frame3.radiobutton3
radiobutton .samplerate.frame3.radiobutton3 -text {Time dependent algorithm} -value {algorithm} -variable {freq_new}

# build widget .samplerate.frame4
frame .samplerate.frame4 -borderwidth {2} -height {30} -relief {ridge} -width {30}

# build widget .samplerate.frame4.button5
button .samplerate.frame4.button5 -background {slategrey} -command {DestroyWindow.samplerate} -font
{-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground {yellow} -padx {11} -pady {4} -text {Done}

# build widget .samplerate.frame4.button6
button .samplerate.frame4.button6 -background {slategrey} -command {setrate} -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} -foreground
{yellow} -padx {11} -pady {4} -text {Apply}

# build widget .samplerate.frame5
frame .samplerate.frame5 -height {30} -width {30}

# build widget .samplerate.frame5.radiobutton9
radiobutton .samplerate.frame5.radiobutton9 -text {1 Hz} -value {1} -variable {freq_new} -width {4}

# build widget .samplerate.frame5.radiobutton10
radiobutton .samplerate.frame5.radiobutton10 -text {2 Hz} -value {2} -variable {freq_new} -width {4}

# build widget .samplerate.frame5.radiobutton11
radiobutton .samplerate.frame5.radiobutton11 -text {3 Hz} -value {3} -variable {freq_new} -width {4}

# build widget .samplerate.frame5.radiobutton12
radiobutton .samplerate.frame5.radiobutton12 -text {4 Hz} -value {4} -variable {freq_new} -width {4}

# build widget .samplerate.frame5.radiobutton13
radiobutton .samplerate.frame5.radiobutton13 -text {5 Hz} -value {5} -variable {freq_new} -width {4}

# pack master .samplerate.frame2
pack configure .samplerate.frame2.radiobutton3 -fill x -side left
pack configure .samplerate.frame2.entry2 -side left

# pack master .samplerate.frame3
pack configure .samplerate.frame3.radiobutton3 -fill x -side left

# pack master .samplerate.frame4
pack configure .samplerate.frame4.button6 -expand 1 -fill x -side left
pack configure .samplerate.frame4.button5 -expand 1 -fill x -side left

# pack master .samplerate.frame5
pack configure .samplerate.frame5.radiobutton9 -expand 1 -fill x -side left
pack configure .samplerate.frame5.radiobutton10 -expand 1 -fill x -side left
pack configure .samplerate.frame5.radiobutton11 -expand 1 -fill x -side left
pack configure .samplerate.frame5.radiobutton12 -expand 1 -fill x -side left
pack configure .samplerate.frame5.radiobutton13 -expand 1 -fill x -side left

# pack master .samplerate
pack configure .samplerate.frame5 -fill x
pack configure .samplerate.frame2 -fill x

```

```

pack configure .samplerate.frame3 -fill x
pack configure .samplerate.frame4 -fill x

if {"[info procs XFEdit]" != ""} {
    catch "XFMiscBindWidgetTree .samplerate"
    after 2 "catch {XFEditSetShowWindows}"
}
}

proc DestroyWindow.samplerate {} {# xf ignore me 7
if {"[info procs XFEdit]" != ""} {
    if {"[info commands .samplerate]" != ""} {
        global xfShowWindow.samplerate
        set xfShowWindow.samplerate 0
        XFEditSetPath .
        after 2 "XFSaveAsProc .samplerate; XFEditSetShowWindows"
    }
} {
    catch "destroy .samplerate"
    update
}
}

# procedure to show window .
proc ShowWindow. {args} {# xf ignore me 7

# Window manager configurations
wm positionfrom . user
wm sizefrom . ""
wm geometry . 681x431
wm maxsize . 1024 768
wm minsize . 425 344
wm protocol . WM_DELETE_WINDOW {XFProcError {Application windows can not be destroyed.
Please use the "Current widget path:" to show/hide windows.}}
wm title . {icsgraph.tcl}

# build widget .frame6
frame .frame6 \
    -borderwidth {2} \
    -height {30} \
    -relief {ridge} \
    -width {30}

# build widget .frame6.button2
button .frame6.button2 \
    -background {slategrey} \
    -command {ShowWindow.samplerate} \
    -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} \
    -foreground {yellow} \
    -highlightthickness {0} \
    -padx {11} \
    -pady {4} \
    -text {Sample Rate}

# build widget .frame6.button4
button .frame6.button4 \
    -background {slategrey} \
    -command {plot_toggle} \
    -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} \
    -foreground {yellow} \
    -highlightthickness {0} \
    -padx {11} \
    -pady {4} \
    -text {Suspend Plotting} \
    -textvariable {plotbutton} \
    -width {11}

# build widget .frame6.button5
button .frame6.button5 \
    -background {slategrey} \
    -command {ics_quit} \
    -font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} \
    -foreground {yellow} \

```

```

-highlightthickness {0} \
-padx {11} \
-pady {4} \
-text {Done}

# build widget .frame6.menubutton0
menubutton .frame6.menubutton0 \
-background {slategrey} \
-font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} \
-foreground {yellow} \
-menu {.frame6.menubutton0.m} \
-padx {5} \
-pady {4} \
-relief {raised} \
-text {File}

# build widget .frame6.menubutton0.m
menu .frame6.menubutton0.m \
-borderwidth {0} \
-tearoff {0}
.frame6.menubutton0.m add command \
-command {param_list_load} \
-label {Load}
.frame6.menubutton0.m add command \
-command {param_list_save} \
-label {Save}

# build widget .frame1
frame .frame1 \
-background {steelblue} \
-borderwidth {2} \
-height {30} \
-relief {ridge} \
-width {30}

# build widget .frame1.frame6
frame .frame1.frame6 \
-borderwidth {2} \
-height {30} \
-relief {ridge} \
-width {30}

# build widget .frame1.frame6.label1
label .frame1.frame6.label1 \
-background {steelblue} \
-borderwidth {0} \
-font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} \
-foreground {white} \
-text {Status:}

# build widget .frame1.frame6.label2
label .frame1.frame6.label2 \
-background {steelblue} \
-borderwidth {0} \
-font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} \
-foreground {green} \
-text {Run} \
-textvariable {status} \
-width {7}

# build widget .frame1.frame7
frame .frame1.frame7 \
-borderwidth {2} \
-height {30} \
-relief {ridge} \
-width {30}

# build widget .frame1.frame7.label4
label .frame1.frame7.label4 \
-background {steelblue} \
-borderwidth {0} \
-font {-adobe-helvetica-medium-r-normal--14-140-75-75-p-77-iso8859-1} \
-foreground {white} \
-text {Simulation time:}

```



```
# build widget .frame1.frame7.label3
label .frame1.frame7.label3 \
  -background {steelblue} \
  -borderwidth {0} \
  -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} \
  -foreground {green} \
  -text {00:07:50} \
  -textvariable {simtime}
```

```
# build widget .frame1.radiobutton0
radiobutton .frame1.radiobutton0 \
  -background {steelblue} \
  -command {showgraphs $totalgraphs} \
  -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} \
  -highlightthickness {0} \
  -text {1} \
  -value {1} \
  -variable {totalgraphs}
```

```
# build widget .frame1.radiobutton1
radiobutton .frame1.radiobutton1 \
  -background {steelblue} \
  -command {showgraphs $totalgraphs} \
  -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} \
  -highlightthickness {0} \
  -text {2} \
  -value {2} \
  -variable {totalgraphs}
```

```
# build widget .frame1.radiobutton2
radiobutton .frame1.radiobutton2 \
  -background {steelblue} \
  -command {showgraphs $totalgraphs} \
  -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} \
  -highlightthickness {0} \
  -text {3} \
  -value {3} \
  -variable {totalgraphs}
```

```
# build widget .frame1.radiobutton3
radiobutton .frame1.radiobutton3 \
  -background {steelblue} \
  -command {showgraphs $totalgraphs} \
  -font {-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1} \
  -highlightthickness {0} \
  -text {4} \
  -value {4} \
  -variable {totalgraphs}
```

```
# build widget .gbase0
frame .gbase0
```

```
# build widget .gbase0.graph0
if {[catch "graph .gbase0.graph0"]} {
  if {"[info procs XFEdit]" != ""} {
    XFPProcError "Unknown widget type: graph"
    return
  }
  puts stderr "Unknown widget type: graph"
  catch "destroy ."
  catch "exit 0"
}
```

```
.gbase0.graph0 config \
  -background {lightgrey} \
  -bufferelements {0} \
  -halo {48} \
  -height {200} \
  -relief {ridge} \
  -title {} \
  -width {200}
```

```
# build widget .gbase0.graph0.label2
label .gbase0.graph0.label2 \
  -background {lightgrey} \
```

```

-borderwidth {0} \
-font {-adobe-times-medium-i-normal--12-120-75-75-p-63-iso8859-1} \
-text {Graph 1}

# build widget .gbase0.graph0.frame18
frame .gbase0.graph0.frame18 \
-background {lightblue} \
-height {30} \
-width {30}

# build widget .gbase0.graph0.frame18.entry14
entry .gbase0.graph0.frame18.entry14 \
-background {lightblue} \
-borderwidth {1} \
-font {-adobe-helvetica-medium-r-normal--10-100-75-75-p-56-iso8859-1} \
-highlightthickness {0} \
-textvariable {param(0)} \
-width {16}
# bindings
bind .gbase0.graph0.frame18.entry14 <Key-Return> {sendparm 0 0}

# build widget .gbase0.graph0.frame18.menubutton0
menubutton .gbase0.graph0.frame18.menubutton0 \
-background {slategrey} \
-font {-adobe-helvetica-medium-r-normal--10-100-75-75-p-56-iso8859-1} \
-foreground {yellow} \
-menu {.gbase0.graph0.frame18.menubutton0.m} \
-padx {0} \
-pady {1} \
-relief {raised} \
-text {1.1732} \
-textvariable {paramval(0)} \
-width {7}

# build widget .gbase0.graph0.frame18.menubutton0.m
menu .gbase0.graph0.frame18.menubutton0.m \
-borderwidth {0} \
-tearoff {0}
.gbase0.graph0.frame18.menubutton0.m add radiobutton \
-label {Display Minimum} \
-value {min} \
-variable {parmdisp(0)}
.gbase0.graph0.frame18.menubutton0.m add radiobutton \
-label {Display Current} \
-value {cur} \
-variable {parmdisp(0)}
.gbase0.graph0.frame18.menubutton0.m add radiobutton \
-label {Display Maximum} \
-value {max} \
-variable {parmdisp(0)}
.gbase0.graph0.frame18.menubutton0.m add separator
.gbase0.graph0.frame18.menubutton0.m add command \
-command {set paramlo(0) $paramcur(0)} \
-label {Reset Minimum}
.gbase0.graph0.frame18.menubutton0.m add command \
-command {set parahi(0) $paramcur(0)} \
-label {Reset Maximum}
.gbase0.graph0.frame18.menubutton0.m add command \
-command {sendparm 0 0} \
-label {Reset Parameter}
.gbase0.graph0.frame18.menubutton0.m add separator
.gbase0.graph0.frame18.menubutton0.m add command \
-command {getparm 0}
wm title .query "Select parameter for Graph 1" \
-label {Select Parameter}
.gbase0.graph0.frame18.menubutton0.m add command \
-command {ShowWindow.range}
wm title .range "Select display range for Graph 1" \
-label {Range adjust}
.gbase0.graph0.frame18.menubutton0.m add separator
.gbase0.graph0.frame18.menubutton0.m add command \
-command {set param(0) ""}
sendparm 0 0 \
-label {Remove Parameter}

```

```

# build widget .gbase0.graph1
if {[catch "graph .gbase0.graph1"]} {
  if {"[info procs XFEdit]" != ""} {
    XFProcError "Unknown widget type: graph"
    return
  }
  puts stderr "Unknown widget type: graph"
  catch "destroy ."
  catch "exit 0"
}
}

.gbase0.graph1 config \
-background {lightgrey} \
-bufferelements {0} \
-halo {48} \
-height {200} \
-relief {ridge} \
-title {} \
-width {200}

# build widget .gbase0.graph1.label2
label .gbase0.graph1.label2 \
-background {lightgrey} \
-borderwidth {0} \
-font {-adobe-times-medium-i-normal-12-120-75-75-p-63-iso8859-1} \
-text {Graph 2}

# build widget .gbase0.graph1.frame18
frame .gbase0.graph1.frame18 \
-background {lightblue} \
-height {30} \
-width {30}

# build widget .gbase0.graph1.frame18.entry14
entry .gbase0.graph1.frame18.entry14 \
-background {lightblue} \
-borderwidth {1} \
-font {-adobe-helvetica-medium-r-normal-10-100-75-75-p-56-iso8859-1} \
-highlightthickness {0} \
-textvariable {param(4)} \
-width {16}

# bindings
bind .gbase0.graph1.frame18.entry14 <Key-Return> {sendparm 1 4}

# build widget .gbase0.graph1.frame18.menubutton0
menubutton .gbase0.graph1.frame18.menubutton0 \
-background {slategrey} \
-font {-adobe-helvetica-medium-r-normal-10-100-75-75-p-56-iso8859-1} \
-foreground {yellow} \
-menu {.gbase0.graph1.frame18.menubutton0.m} \
-padx {0} \
-pady {1} \
-relief {raised} \
-textvariable {parmval(4)} \
-width {7}

# build widget .gbase0.graph1.frame18.menubutton0.m
menu .gbase0.graph1.frame18.menubutton0.m \
-borderwidth {0} \
-learoff {0}
.gbase0.graph1.frame18.menubutton0.m add radiobutton \
-label {Display Minimum} \
-value {min} \
-variable {parmdisp(4)}
.gbase0.graph1.frame18.menubutton0.m add radiobutton \
-label {Display Current} \
-value {cur} \
-variable {parmdisp(4)}
.gbase0.graph1.frame18.menubutton0.m add radiobutton \
-label {Display Maximum} \
-value {max} \
-variable {parmdisp(4)}
.gbase0.graph1.frame18.menubutton0.m add separator
.gbase0.graph1.frame18.menubutton0.m add command \
-command {set parmlo(4) $parmc(4)} \

```

```

-label {Reset Minimum}
.gbase0.graph1.frame18.menubutton0.m add command \
-command {set parmhi(4) $parmcu(4)} \
-label {Reset Maximum}
.gbase0.graph1.frame18.menubutton0.m add command \
-command {sendparm 1 4} \
-label {Reset Parameter}
.gbase0.graph1.frame18.menubutton0.m add separator
.gbase0.graph1.frame18.menubutton0.m add command \
-command {getparm 4}
wm title .query "Select parameter for Graph 2" \
-label {Select Parameter}
.gbase0.graph1.frame18.menubutton0.m add command \
-command {ShowWindow.range}
wm title .range "Select display range for Graph 2" \
-label {Range adjust}
.gbase0.graph1.frame18.menubutton0.m add separator
.gbase0.graph1.frame18.menubutton0.m add command \
-command {set param(4) ""}
sendparm 1 4 \
-label {Remove Parameter}

# build widget .gbase0.frame0
frame .gbase0.frame0 \
-borderwidth {2} \
-height {30} \
-relief {ridge} \
-width {30}

# build widget .gbase0.frame0.frame18
frame .gbase0.frame0.frame18 \
-background {lightblue} \
-height {30} \
-width {30}

# build widget .gbase0.frame0.frame18.entry14
entry .gbase0.frame0.frame18.entry14 \
-background {lightblue} \
-borderwidth {1} \
-font {-adobe-helvetica-medium-r-normal--10-100-75-75-p-56-iso8859-1} \
-highlighthickness {0} \
-textvariable {param(3)} \
-width {16}
# bindings
bind .gbase0.frame0.frame18.entry14 <Key-Return> {sendparm 0 3}

# build widget .gbase0.frame0.frame18.menubutton0
menubutton .gbase0.frame0.frame18.menubutton0 \
-background {slategrey} \
-font {-adobe-helvetica-medium-r-normal--10-100-75-75-p-56-iso8859-1} \
-foreground {yellow} \
-menu { .gbase0.frame0.frame18.menubutton0.m } \
-padx {0} \
-pady {1} \
-relief {raised} \
-textvariable {parmval(3)} \
-width {7}

# build widget .gbase0.frame0.frame18.menubutton0.m
menu .gbase0.frame0.frame18.menubutton0.m \
-borderwidth {0} \
-tearoff {0}
.gbase0.frame0.frame18.menubutton0.m add radiobutton \
-label {Display Minimum} \
-value {min} \
-variable {parmdisp(3)}
.gbase0.frame0.frame18.menubutton0.m add radiobutton \
-label {Display Current} \
-value {cu} \
-variable {parmdisp(3)}
.gbase0.frame0.frame18.menubutton0.m add radiobutton \
-label {Display Maximum} \
-value {max} \
-variable {parmdisp(3)}
.gbase0.frame0.frame18.menubutton0.m add separator

```

```

.gbase0.frame0.frame18.menubutton0.m add command \
-command {set parmlo(3) $parmc(3)} \
-label {Reset Minimum}
.gbase0.frame0.frame18.menubutton0.m add command \
-command {set parmhi(3) $parmc(3)} \
-label {Reset Maximum}
.gbase0.frame0.frame18.menubutton0.m add command \
-command {sendparm 0 3} \
-label {Reset Parameter}
.gbase0.frame0.frame18.menubutton0.m add separator
.gbase0.frame0.frame18.menubutton0.m add command \
-command {getparm 3}
wm title .query "Select parameter for Graph 1" \
-label {Select Parameter}
.gbase0.frame0.frame18.menubutton0.m add command \
-command {ShowWindow.range}
wm title .range "Select display range for Graph 1" \
-label {Range adjust}
.gbase0.frame0.frame18.menubutton0.m add separator
.gbase0.frame0.frame18.menubutton0.m add command \
-command {set param(3) ""}
sendparm 0 3 \
-label {Remove Parameter}

# build widget .gbase0.frame0.frame2
frame .gbase0.frame0.frame2 \
-background {lightblue} \
-height {30} \
-width {30}

# build widget .gbase0.frame0.frame2.entry14
entry .gbase0.frame0.frame2.entry14 \
-background {lightblue} \
-borderwidth {1} \
-font {-adobe-helvetica-medium-r-normal--10-100-75-75-p-56-iso8859-1} \
-highlighthickness {0} \
-textvariable {param(2)} \
-width {16}
# bindings
bind .gbase0.frame0.frame2.entry14 <Key-Return> {sendparm 0 2}

# build widget .gbase0.frame0.frame2.menubutton0
menubutton .gbase0.frame0.frame2.menubutton0 \
-background {slategrey} \
-font {-adobe-helvetica-medium-r-normal--10-100-75-75-p-56-iso8859-1} \
-foreground {yellow} \
-menu {.gbase0.frame0.frame2.menubutton0.m} \
-padx {0} \
-pady {1} \
-relief {raised} \
-textvariable {parmval(2)} \
-width {7}

# build widget .gbase0.frame0.frame2.menubutton0.m
menu .gbase0.frame0.frame2.menubutton0.m \
-borderwidth {0} \
-tearoff {0}
.gbase0.frame0.frame2.menubutton0.m add radiobutton \
-label {Display Minimum} \
-value {min} \
-variable {parmdisp(2)}
.gbase0.frame0.frame2.menubutton0.m add radiobutton \
-label {Display Current} \
-value {cur} \
-variable {parmdisp(2)}
.gbase0.frame0.frame2.menubutton0.m add radiobutton \
-label {Display Maximum} \
-value {max} \
-variable {parmdisp(2)}
.gbase0.frame0.frame2.menubutton0.m add separator
.gbase0.frame0.frame2.menubutton0.m add command \
-command {set parmlo(2) $parmc(2)} \
-label {Reset Minimum}
.gbase0.frame0.frame2.menubutton0.m add command \
-command {set parmhi(2) $parmc(2)} \

```

```

-label {Reset Maximum}
.gbase0.frame0.frame2.menubutton0.m add command \
-command {sendparm 0 2} \
-label {Reset Parameter}
.gbase0.frame0.frame2.menubutton0.m add separator
.gbase0.frame0.frame2.menubutton0.m add command \
-command {getparm 2
wm title .query "Select parameter for Graph 1"} \
-label {Select Parameter}
.gbase0.frame0.frame2.menubutton0.m add command \
-command {ShowWindow.range
wm title .range "Select display range for Graph 1"} \
-label {Range adjust}
.gbase0.frame0.frame2.menubutton0.m add separator
.gbase0.frame0.frame2.menubutton0.m add command \
-command {set param(2) ""
sendparm 0 2} \
-label {Remove Parameter}

# build widget .gbase0.frame0.frame4
frame .gbase0.frame0.frame4 \
-background {lightblue} \
-height {30} \
-width {30}

# build widget .gbase0.frame0.frame4.entry14
entry .gbase0.frame0.frame4.entry14 \
-background {lightblue} \
-borderwidth {1} \
-font {-adobe-helvetica-medium-r-normal--10-100-75-75-p-56-iso8859-1} \
-highlightthickness {0} \
-textvariable {param(1)} \
-width {16}
# bindings
bind .gbase0.frame0.frame4.entry14 <Key-Return> {sendparm 0 1}

# build widget .gbase0.frame0.frame4.menubutton0
menubutton .gbase0.frame0.frame4.menubutton0 \
-background {slategrey} \
-font {-adobe-helvetica-medium-r-normal--10-100-75-75-p-56-iso8859-1} \
-foreground {yellow} \
-menu { .gbase0.frame0.frame4.menubutton0.m } \
-padx {0} \
-pady {1} \
-relief {raised} \
-text {17} \
-textvariable {parmval(1)} \
-width {7}

# build widget .gbase0.frame0.frame4.menubutton0.m
menu .gbase0.frame0.frame4.menubutton0.m \
-borderwidth {0} \
-tearoff {0}
.gbase0.frame0.frame4.menubutton0.m add radiobutton \
-label {Display Minimum} \
-value {min} \
-variable {parmdisp(1)}
.gbase0.frame0.frame4.menubutton0.m add radiobutton \
-label {Display Current} \
-value {cur} \
-variable {parmdisp(1)}
.gbase0.frame0.frame4.menubutton0.m add radiobutton \
-label {Display Maximum} \
-value {max} \
-variable {parmdisp(1)}
.gbase0.frame0.frame4.menubutton0.m add separator
.gbase0.frame0.frame4.menubutton0.m add command \
-command {set parmlo(1) $parmcur(1)} \
-label {Reset Minimum}
.gbase0.frame0.frame4.menubutton0.m add command \
-command {set parmhi(1) $parmcur(1)} \
-label {Reset Maximum}
.gbase0.frame0.frame4.menubutton0.m add command \
-command {sendparm 0 1} \
-label {Reset Parameter}

```

```

.gbase0.frame0.frame4.menubutton0.m add separator
.gbase0.frame0.frame4.menubutton0.m add command \
-command {getparm 1}
wm title .query "Select parameter for Graph 1" \
-label {Select Parameter}
.gbase0.frame0.frame4.menubutton0.m add command \
-command {ShowWindow.range}
wm title .range "Select display range for Graph 1" \
-label {Range adjust}
.gbase0.frame0.frame4.menubutton0.m add separator
.gbase0.frame0.frame4.menubutton0.m add command \
-command {set param(1) ""}
sendparm 0 1 \
-label {Remove Parameter}

# build widget .gbase0.frame0.frame6
frame .gbase0.frame0.frame6 \
-background {lightblue} \
-height {30} \
-width {30}

# build widget .gbase0.frame0.frame6.entry14
entry .gbase0.frame0.frame6.entry14 \
-background {lightblue} \
-borderwidth {1} \
-font {-adobe-helvetica-medium-r-normal-10-100-75-75-p-56-iso8859-1} \
-highlightthickness {0} \
-textvariable {param(0)} \
-width {16}
# bindings
bind .gbase0.frame0.frame6.entry14 <Key-Return> {sendparm 0 0}

# build widget .gbase0.frame0.frame6.menubutton0
menubutton .gbase0.frame0.frame6.menubutton0 \
-background {#f08090} \
-font {-adobe-helvetica-medium-r-normal-10-100-75-75-p-56-iso8859-1} \
-foreground {yellow} \
-menu {.gbase0.frame0.frame6.menubutton0.m} \
-padx {0} \
-pady {1} \
-relief {raised} \
-text {1.1732} \
-textvariable {paramval(0)} \
-width {7}

# build widget .gbase0.frame0.frame6.menubutton0.m
menu .gbase0.frame0.frame6.menubutton0.m \
-borderwidth {0} \
-tearoff {0}
.gbase0.frame0.frame6.menubutton0.m add radiobutton \
-label {Display Minimum} \
-value {min} \
-variable {parmdisp(0)}
.gbase0.frame0.frame6.menubutton0.m add radiobutton \
-label {Display Current} \
-value {cur} \
-variable {parmdisp(0)}
.gbase0.frame0.frame6.menubutton0.m add radiobutton \
-label {Display Maximum} \
-value {max} \
-variable {parmdisp(0)}
.gbase0.frame0.frame6.menubutton0.m add separator
.gbase0.frame0.frame6.menubutton0.m add command \
-command {set paramlo(0) $paramcur(0)} \
-label {Reset Minimum}
.gbase0.frame0.frame6.menubutton0.m add command \
-command {set paramhi(0) $paramcur(0)} \
-label {Reset Maximum}
.gbase0.frame0.frame6.menubutton0.m add command \
-command {sendparm 0 0} \
-label {Reset Parameter}
.gbase0.frame0.frame6.menubutton0.m add separator
.gbase0.frame0.frame6.menubutton0.m add command \
-command {getparm 0}
wm title .query "Select parameter for Graph 1" \

```

```

-label {Select Parameter}
.gbase0.frame0.frame6.menubutton0.m add command \
-command {ShowWindow.range
wm title .range "Select display range for Graph 1"} \
-label {Range adjust}
.gbase0.frame0.frame6.menubutton0.m add separator
.gbase0.frame0.frame6.menubutton0.m add command \
-command {set param(0) ""
sendparm 0 0} \
-label {Remove Parameter}

# build widget .gbase1
frame .gbase1

# build widget .gbase1.graph0
if {[catch "graph .gbase1.graph0"]} {
if {"[info procs XFEdit]" != ""} {
XFPProcError "Unknown widget type: graph"
return
} {
puts stderr "Unknown widget type: graph"
catch "destroy ."
catch "exit 0"
}
}
.gbase1.graph0 config \
-background {lightgrey} \
-bufferelements {0} \
-halo {48} \
-height {200} \
-relief {ridge} \
-title {} \
-width {200}

# build widget .gbase1.graph0.label2
label .gbase1.graph0.label2 \
-background {lightgrey} \
-borderwidth {0} \
-font {-adobe-limes-medium-i-normal-12-120-75-75-p-63-iso8859-1} \
-text {Graph 3}

# build widget .gbase1.graph0.frame18
frame .gbase1.graph0.frame18 \
-background {lightblue} \
-height {30} \
-width {30}

# build widget .gbase1.graph0.frame18.entry14
entry .gbase1.graph0.frame18.entry14 \
-background {lightblue} \
-borderwidth {1} \
-font {-adobe-helvetica-medium-r-normal-10-100-75-75-p-56-iso8859-1} \
-highlightthickness {0} \
-textvariable {param(8)} \
-width {16}
# bindings
bind .gbase1.graph0.frame18.entry14 <Key-Return> {sendparm 2 8}

# build widget .gbase1.graph0.frame18.menubutton0
menubutton .gbase1.graph0.frame18.menubutton0 \
-background {slategrey} \
-font {-adobe-helvetica-medium-r-normal-10-100-75-75-p-56-iso8859-1} \
-foreground {yellow} \
-menu {.gbase1.graph0.frame18.menubutton0.m} \
-padx {0} \
-pady {1} \
-relief {raised} \
-textvariable {paramval(8)} \
-width {7}

# build widget .gbase1.graph0.frame18.menubutton0.m
menu .gbase1.graph0.frame18.menubutton0.m \
-borderwidth {0} \
-tearoff {0}
.gbase1.graph0.frame18.menubutton0.m add radiobutton \

```



```

-label {Display Minimum} \
-value {min} \
-variable {parmdisp(8)}
.gbase1.graph0.frame18.menubutton0.m add radiobutton \
-label {Display Current} \
-value {cur} \
-variable {parmdisp(8)}
.gbase1.graph0.frame18.menubutton0.m add radiobutton \
-label {Display Maximum} \
-value {max} \
-variable {parmdisp(8)}
.gbase1.graph0.frame18.menubutton0.m add separator
.gbase1.graph0.frame18.menubutton0.m add command \
-command {set parml(8) $parmcur(8)} \
-label {Reset Minimum}
.gbase1.graph0.frame18.menubutton0.m add command \
-command {set parmhi(8) $parmcur(8)} \
-label {Reset Maximum}
.gbase1.graph0.frame18.menubutton0.m add command \
-command {sendparm 2 8} \
-label {Reset Parameter}
.gbase1.graph0.frame18.menubutton0.m add separator
.gbase1.graph0.frame18.menubutton0.m add command \
-command {getparm 8}
wm title .query "Select parameter for Graph 3" \
-label {Select Parameter}
.gbase1.graph0.frame18.menubutton0.m add command \
-command {ShowWindow.range}
wm title .range "Select display range for Graph 3" \
-label {Range adjust}
.gbase1.graph0.frame18.menubutton0.m add separator
.gbase1.graph0.frame18.menubutton0.m add command \
-command {set param(8) ""}
sendparm 2 8 \
-label {Remove Parameter}

# build widget .gbase1.graph2
if {[catch "graph .gbase1.graph2"]} {
if {"[info procs XFEdit]" != ""} {
XFProcError "Unknown widget type: graph"
return
}
puts stderr "Unknown widget type: graph"
catch "destroy ."
catch "exit 0"
}
}

.gbase1.graph2 config \
-background {lightgrey} \
-bufferelements {0} \
-halo {48} \
-height {200} \
-relief {ridge} \
-title {} \
-width {200}

# build widget .gbase1.graph2.label2
label .gbase1.graph2.label2 \
-background {lightgrey} \
-borderwidth {0} \
-font {-adobe-times-medium-i-normal--12-120-75-75-p-63-iso8859-1} \
-text {Graph 4}

# build widget .gbase1.graph2.frame18
frame .gbase1.graph2.frame18 \
-background {lightblue} \
-height {30} \
-width {30}

# build widget .gbase1.graph2.frame18.entry14
entry .gbase1.graph2.frame18.entry14 \
-background {lightblue} \
-borderwidth {1} \
-font {-adobe-helvetica-medium-r-normal--10-100-75-75-p-56-iso8859-1} \
-highlightthickness {0} \

```

```

-textvariable {param(12)} \
-width {16}
# bindings
bind .gbase1.graph2.frame18.entry14 <Key-Return> {sendparm 3 12}

# build widget .gbase1.graph2.frame18.menubutton0
menubutton .gbase1.graph2.frame18.menubutton0 \
-background {slategrey} \
-font {-adobe-helvetica-medium-r-normal--10-100-75-75-p-56-iso8859-1} \
-foreground {yellow} \
-menu {.gbase1.graph2.frame18.menubutton0.m} \
-padx {0} \
-pady {1} \
-relief {raised} \
-textvariable {paramval(12)} \
-width {7}

# build widget .gbase1.graph2.frame18.menubutton0.m
menu .gbase1.graph2.frame18.menubutton0.m \
-borderwidth {0} \
-tearoff {0}
.gbase1.graph2.frame18.menubutton0.m add radiobutton \
-label {Display Minimum} \
-value {min} \
-variable {parmdisp(12)}
.gbase1.graph2.frame18.menubutton0.m add radiobutton \
-label {Display Current} \
-value {cur} \
-variable {parmdisp(12)}
.gbase1.graph2.frame18.menubutton0.m add radiobutton \
-label {Display Maximum} \
-value {max} \
-variable {parmdisp(12)}
.gbase1.graph2.frame18.menubutton0.m add separator
.gbase1.graph2.frame18.menubutton0.m add command \
-command {set paramlo(12) $paramcur(12)} \
-label {Reset Minimum}
.gbase1.graph2.frame18.menubutton0.m add command \
-command {set paramhi(8) $paramcur(8)} \
-label {Reset Maximum}
.gbase1.graph2.frame18.menubutton0.m add command \
-command {sendparm 3 12} \
-label {Reset Parameter}
.gbase1.graph2.frame18.menubutton0.m add separator
.gbase1.graph2.frame18.menubutton0.m add command \
-command {getparm 12}
wm title .query "Select parameter for Graph 4" \
-label {Select Parameter}
.gbase1.graph2.frame18.menubutton0.m add command \
-command {ShowWindow.range}
wm title .range "Select display range for Graph 4" \
-label {Range adjust}
.gbase1.graph2.frame18.menubutton0.m add separator
.gbase1.graph2.frame18.menubutton0.m add command \
-command {set param(12) ""}
sendparm 3 12} \
-label {Remove Parameter}

# pack master .frame6
pack configure .frame6.button2 \
-expand 1 \
-fill x \
-side left
pack configure .frame6.button4 \
-expand 1 \
-fill x \
-side left
pack configure .frame6.menubutton0 \
-expand 1 \
-fill x \
-side left
pack configure .frame6.button5 \
-expand 1 \
-fill x \
-side left

```

```
# pack master .frame1
pack configure .frame1.frame6 \
-side left
pack configure .frame1.frame7 \
-side left
pack configure .frame1.radiobutton3 \
-side right
pack configure .frame1.radiobutton2 \
-side right
pack configure .frame1.radiobutton1 \
-side right
pack configure .frame1.radiobutton0 \
-side right

# pack master .frame1.frame6
pack configure .frame1.frame6.label1 \
-side left
pack configure .frame1.frame6.label2 \
-fill both \
-side left

# pack master .frame1.frame7
pack configure .frame1.frame7.label4 \
-side left
pack configure .frame1.frame7.label3 \
-fill both \
-side left

# pack master .gbase0
pack configure .gbase0.graph0 \
-expand 1 \
-fill both
pack configure .gbase0.frame0 \
-anchor w

# pack master .gbase0.graph0
pack configure .gbase0.graph0.label2 \
-anchor w

# pack master .gbase0.graph0.frame18
pack configure .gbase0.graph0.frame18.menubutton0 \
-side left
pack configure .gbase0.graph0.frame18.entry14 \
-anchor w \
-side bottom

# pack master .gbase0.graph1
pack configure .gbase0.graph1.label2 \
-anchor w \
-fill y
pack configure .gbase0.graph1.frame18 \
-anchor w \
-side bottom

# pack master .gbase0.graph1.frame18
pack configure .gbase0.graph1.frame18.menubutton0 \
-side left
pack configure .gbase0.graph1.frame18.entry14 \
-side left

# pack master .gbase0.frame0
pack configure .gbase0.frame0.frame6 \
-anchor w \
-side left
pack configure .gbase0.frame0.frame4 \
-anchor w \
-side left
pack configure .gbase0.frame0.frame2 \
-anchor w \
-side left
pack configure .gbase0.frame0.frame18 \
-anchor w \
-side left
```

```
# pack master .gbase0.frame0.frame18
pack configure .gbase0.frame0.frame18.menubutton0 \
-side left
pack configure .gbase0.frame0.frame18.entry14 \
-side left
```

```
# pack master .gbase0.frame0.frame2
pack configure .gbase0.frame0.frame2.menubutton0 \
-side left
pack configure .gbase0.frame0.frame2.entry14 \
-side left
```

```
# pack master .gbase0.frame0.frame4
pack configure .gbase0.frame0.frame4.menubutton0 \
-side left
pack configure .gbase0.frame0.frame4.entry14 \
-side left
```

```
# pack master .gbase0.frame0.frame6
pack configure .gbase0.frame0.frame6.menubutton0 \
-side left
pack configure .gbase0.frame0.frame6.entry14 \
-side left
```

```
# pack master .gbase1
pack configure .gbase1.graph0 \
-expand 1 \
-fill both
```

```
# pack master .gbase1.graph0
pack configure .gbase1.graph0.label2 \
-anchor w
pack configure .gbase1.graph0.frame18 \
-anchor w \
-side bottom
```

```
# pack master .gbase1.graph0.frame18
pack configure .gbase1.graph0.frame18.menubutton0 \
-side left
pack configure .gbase1.graph0.frame18.entry14 \
-side left
```

```
# pack master .gbase1.graph2
pack configure .gbase1.graph2.label2 \
-anchor w
pack configure .gbase1.graph2.frame18 \
-anchor w \
-side bottom
```

```
# pack master .gbase1.graph2.frame18
pack configure .gbase1.graph2.frame18.menubutton0 \
-side left
pack configure .gbase1.graph2.frame18.entry14 \
-side left
```

```
# pack master .
pack configure .frame1 \
-fill x
pack configure .gbase0 \
-expand 1 \
-fill both
pack configure .frame6 \
-fill x \
-side bottom
```

```
.gbase0.graph0 configure -background lightgrey -barmode normal -barwidth 0.95 -borderwidth 2 -bottommargin 0 -bufferelements 0 -cursor crosshair -font
*Helvetica-Bold-R-Normal*-14-140-* -foreground black -halo 48 -height 200 -inverby 0 -justify center -leftmargin 0 -plotbackground white -plotborderwidth 2 -plotrelief
sunken -relief ridge -rightmargin 0 -takefocus {} -tile {} -title {} -topmargin 0 -width 200
.gbase0.graph0 element create line0 -activecolor blue -activefill defcolor -activelinewidth 1 -activeoutline defcolor -color black -dashes {} -fill defcolor -label edibattery
-linewidth 1 -mapped 1 -mapx x -mapy y -offdash {} -outline defcolor -outlinewidth 1 -pixels 12 -scalesymbols 0 -symbol none -trace both -xdata xval0 -ydata yval0
.gbase0.graph0 element create line1 -activecolor blue -activefill defcolor -activelinewidth 1 -activeoutline defcolor -color blue -dashes {} -fill defcolor -label edivoltreg
-linewidth 1 -mapped 1 -mapx x -mapy y -offdash {} -outline defcolor -outlinewidth 1 -pixels 12 -scalesymbols 0 -symbol none -trace both -xdata xval1 -ydata yval1
.gbase0.graph0 element create line2 -activecolor blue -activefill defcolor -activelinewidth 1 -activeoutline defcolor -color red -dashes {} -data {} -fill defcolor -label edravgcell
-linewidth 1 -mapped 1 -mapx x -mapy y -offdash {} -outline defcolor -outlinewidth 1 -pixels 12 -scalesymbols 0 -symbol none -trace both -xdata xval2 -ydata yval2
```



```
.gbase1.graph2 element create line15 -activecolor blue -activefill defcolor -activelinewidth 1 -activeoutline defcolor -color orange -dashes {} -data {} -fill defcolor -label {}
-linewidth 1 -mapped 1 -mapx x -mapy y -offdash {} -outline defcolor -outlinewidth 1 -pixels 12 -scalesymbols 0 -symbol none -trace both -xdata xval15 -ydata yval15
.gbase1.graph2 axis configure -color black -command {} -descending 0 -justify center -linewidth 0 -logscale 0 -loose 0 -majorticks {} -mapped 1 -max {} -min 0.0 -minorticks
{} -rotate 0.0 -showticks 1 -stepsize 0.0 -subdivisions 2 -tickfont "Courier-Bold-R-Normal"-100- -ticklength 10 -title {Time (secs)} -titlefont
"Helvetica-Bold-R-Normal"-12-120-
.gbase1.graph2 yaxis configure -color black -command {} -descending 0 -justify center -linewidth 0 -logscale 0 -loose 0 -majorticks {} -mapped 1 -max {} -min {} -minorticks
{} -rotate 0.0 -showticks 1 -stepsize 0.0 -subdivisions 2 -tickfont "Courier-Bold-R-Normal"-100- -ticklength 10 -title {} -titlefont "Helvetica-Bold-R-Normal"-12-120-
.gbase1.graph2 x2axis configure -color black -command {} -descending 0 -justify center -linewidth 0 -logscale 0 -loose 0 -majorticks {} -mapped 0 -max {} -min {} -minorticks
{} -rotate 0.0 -showticks 0 -stepsize 0.0 -subdivisions 2 -tickfont "Courier-Bold-R-Normal"-100- -ticklength 10 -title X2 -titlefont "Helvetica-Bold-R-Normal"-12-120-
.gbase1.graph2 y2axis configure -color black -command {} -descending 0 -justify center -linewidth 0 -logscale 0 -loose 0 -majorticks {} -mapped 0 -max {} -min {} -minorticks
{} -rotate 0.0 -showticks 1 -stepsize 0.0 -subdivisions 2 -tickfont "Courier-Bold-R-Normal"-100- -ticklength 10 -title Y2 -titlefont "Helvetica-Bold-R-Normal"-12-120-
.gbase1.graph2 legend configure -activebackground systembuttonhighlight -activeborderwidth 2 -activeforeground black -activerelief flat -anchor n -background {}
-borderwidth 2 -font "Helvetica-Bold-R-Normal"-10-100- -foreground black -ipadx 1 -ipady 1 -mapped 0 -padx {4 4} -pady {4 4} -position rightmargin -raised 0 -relief sunken
.gbase1.graph2 crosshairs configure -color black -dashes {} -linewidth 0 -mapped 0 -position @0,0
.gbase1.graph2 postscript configure -center 1 -colormap {} -colormode color -decorations 1 -fontmap {} -height 0 -landscape 0 -maxpct 0 -padx {96 96} -pady {96 96}
-paperheight 1059 -paperwidth 818 -width 0
```

```
if {[info procs XFEdit] != ""} {
  catch "XFMiscBindWidgetTree ."
  after 2 "catch [XFEditSetShowWindows]"
}
}
```

```
# User defined procedures
```

```
# Procedure: css_stat
proc css_stat { cmd } {
  css_cmd stat $cmd
}
```

```
proc plot_toggle {} {
# Toggle the plotting of data points ON/OFF
#
global plotting
global plotbutton
global plotbuf_x
global plotbuf_y
global xval0
global xval1
global xval2
global xval3
global xval4
global xval5
global xval6
global xval7
global xval8
global xval9
global xval10
global xval11
global xval12
global xval13
global xval14
global xval15
global yval0
global yval1
global yval2
global yval3
global yval4
global yval5
global yval6
global yval7
global yval8
global yval9
global yval10
global yval11
global yval12
global yval13
global yval14
global yval15
```

```
if {$plotting} {
  set plotbutton "Resume Plotting"
```

```

set plotting 0
for {set ctr 0} {$ctr < 16} {incr ctr} {
    set plotbuf_x($ctr) ""
    set plotbuf_y($ctr) ""
}
{
    set plotbutton "Suspend Plotting"
    set plotting 1
    for {set ctr 0} {$ctr < 16} {incr ctr} {
        foreach item_x $plotbuf_x($ctr) item_y $plotbuf_y($ctr) {
            set xval${ctr}{++end} $item_x
            set yval${ctr}{++end} $item_y
        }
    }
}
}

```

Procedure: cyclic_graph

```

proc cyclic_graph {} {
    global clockstr
    global datesr
    global status
    global simtime
    global simsecs
    global cpusage
    global curlC
    global dataListid
    global parmval
    global DataSrcList
    global xval0
    global xval1
    global xval2
    global xval3
    global xval4
    global xval5
    global xval6
    global xval7
    global xval8
    global xval9
    global xval10
    global xval11
    global xval12
    global xval13
    global xval14
    global xval15
    global yval0
    global yval1
    global yval2
    global yval3
    global yval4
    global yval5
    global yval6
    global yval7
    global yval8
    global yval9
    global yval10
    global yval11
    global yval12
    global yval13
    global yval14
    global yval15
    global simsecslast
    global parmlo
    global parmhi
    global parmdisp
    global parmcur
    global dispmode
    global tailsze
    global mirval
    global maxval
    global plotbuf_x
    global plotbuf_y
    global plotting
    global simrate
    global load_name
}

```

```

global cyclicrate
global cyclic_on
global baserate
global freq
global cyclic_on
global graphlist

if {$cyclic_on == "0"} {
  after $cyclicrate {cyclic_graph}
  return
}
getval_cyclic
set clockstr [clock format [clock seconds] -format %X]
set datestr [clock format [clock seconds] -format %x]
set info [css_cmd ctrl status]
set status [lindex $info 0]
set rate [lindex $info 1]
set simrate $rate
if {$simrate == 0.00 || $simrate == ""} {set simrate 1.0}
set cyclicrate [expr int(($baserate / $freq) / $simrate)]
set ratestr "Rate: $rate"
set simtime [lindex $info 2]
set simsecs [lindex $info 3]
set cpusage [lindex $info 4]
set loadname [file tail [lindex $info 5]]
set curlC [lindex $info 6]
if {$simsecs < $simsecslast} {
  sendparm 0 all; setranges 0
  sendparm 1 all; setranges 1
  sendparm 2 all; setranges 2
  sendparm 3 all; setranges 3
}
set simsecslast $simsecs
if {$status == "Run"} {
  for {set gnum 0} {$gnum < 4} {incr gnum} {
    if {$sdispmode($gnum) == "tail" && $plotting} {
      if {$simsecs > $maxval($gnum)} {
        set minval [expr $simsecs - $tailsz($gnum) / 2]
        if {$minval < 0} {
          set minval 0
        }
        set maxval($gnum) [expr $minval + $tailsz($gnum)]
        [lindex $graphlist $gnum] xaxis configure -min $minval -max $maxval($gnum)
      }
    }
  }
  set vlist [css_cmd stat "getlist $datalistid"]
  set ctr 0
  foreach item $DataSrcList {
    if {$item == "a_z"} {
      set parmval($ctr) ""
    }
    {
      set pval [lindex $vlist $ctr]
      set parmcur($ctr) $pval
      if {$plotting} {
        set xval($ctr){++end} $simsecs
        set yval($ctr){++end} $pval
      }
      lappend plotbuf_x($ctr) $simsecs
      lappend plotbuf_y($ctr) $pval
    }
    if {$pval > $parmhi($ctr)} {set parmhi($ctr) $pval}
    if {$pval < $parmlo($ctr)} {set parmlo($ctr) $pval}
    switch $parmdisp($ctr) {
      min {set parmval($ctr) $parmlo($ctr)}
      cur {set parmval($ctr) $pval}
      max {set parmval($ctr) $parmhi($ctr)}
    }
  }
  incr ctr
}
}
after $cyclicrate {cyclic_graph}
}

```



```

# Procedure: getparam
proc getparam { param} {
# This procedure provides a list box for selecting
# a display parameter.
#
global curparam

set curparam $param
ShowWindow.query
}

```

```

# Procedure: graph_config
proc graph_config {} {
global xval0
global xval1
global xval2
global xval3
global xval4
global xval5
global xval6
global xval7
global xval8
global xval9
global xval10
global xval11
global xval12
global xval13
global xval14
global xval15
global yval0
global yval1
global yval2
global yval3
global yval4
global yval5
global yval6
global yval7
global yval8
global yval9
global yval10
global yval11
global yval12
global yval13
global yval14
global yval15

```

```

vector xval0(1) xval1(1) xval2(1) xval3(1)
vector xval4(1) xval5(1) xval6(1) xval7(1)
vector xval8(1) xval9(1) xval10(1) xval11(1)
vector xval12(1) xval13(1) xval14(1) xval15(1)
vector yval0(1) yval1(1) yval2(1) yval3(1)
vector yval4(1) yval5(1) yval6(1) yval7(1)
vector yval8(1) yval9(1) yval10(1) yval11(1)
vector yval12(1) yval13(1) yval14(1) yval15(1)
}

```

```

# Procedure: param_list_load
proc param_list_load {} {
# This procedure retrieves the current list of graphed parameters from file
#
global DataSrcList
global param
global tailsize
global dispmode
global yscalemode
global baserate
global cyclicrate
global freq

set types {
    {"ICS graph files"      { .grf      }
    {"Text files"          { .txt      }
}

```

```

        {"All files"      ""}
    }
set result [tk_getOpenFile -filetypes $types -initialdir ./graph -title "Load Graph Parameters" -defaultextension .grf]
if {$result != ""} {
    set fin [open $result r]
    gets $fin line
    if {$line == "# Graph parameters"} {
        while (![eof $fin]) {
            gets $fin line
            set arg0 [lindex $line 0]
            set arg1 [lindex $line 1]
            set arg2 [lindex $line 2]
            switch $arg0 {
                param          {set param($arg1) $arg2}
                tailsize       {set tailsize($arg1) $arg2}
                dispmode       {set dispmode($arg1) $arg2}
                yscalemode     {set yscalemode($arg1) $arg2}
                freq            {set freq $arg1}
            }
        }
        sendparam 0 all
        sendparam 1 all
        sendparam 2 all
        sendparam 3 all
        setranges 0
        setranges 1
        setranges 2
        setranges 3
    }
    close $fin
}
}

```

Procedure: param_list_save

```

proc param_list_save {} {
# This procedure saves the current list of graphed parameters to file
#
global DataSrcList
global param
global tailsize
global dispmode
global yscalemode
global baserate
global cyclicrate
global freq

set types {
    {"ICS graph files"  { .grf } }
    {"Text files"      { .txt } }
    {"All files"       ""}
}

```

```

set result [tk_getSaveFile -filetypes $types -initialdir ./graph -title "Save Graph Parameters" -initialfile Untitled -defaultextension .grf]
if {$result != ""} {
    set fout [open $result w]
    puts $fout "# Graph parameters"
    for {set ctr 0} {$ctr < 4} {incr ctr} {
        set pnum [expr $ctr * 4]
        puts $fout "param $ctr $param($pnum)"
        puts $fout "tailsize $ctr $tailsize($ctr)"
        puts $fout "dispmode $ctr $dispmode($ctr)"
        puts $fout "yscalemode $ctr $yscalemode($ctr)"
    }
    puts $fout "param 1 $param(1)"
    puts $fout "param 2 $param(2)"
    puts $fout "param 3 $param(3)"
    puts $fout "freq $freq"
    close $fout
}
}

```

Procedure: sendparam

```

proc sendparam { gnum pnum } {
# This procedure creates a new DataSrcList and sends it

```

```

# to the server.
#
# gnum = sequential graph identifier (1-4)
# pnum = parameter number for the graph
#
global DataSrcList
global datalistid
global param
global xval0
global xval1
global xval2
global xval3
global xval4
global xval5
global xval6
global xval7
global xval8
global xval9
global xval10
global xval11
global xval12
global xval13
global xval14
global xval15
global yval0
global yval1
global yval2
global yval3
global yval4
global yval5
global yval6
global yval7
global yval8
global yval9
global yval10
global yval11
global yval12
global yval13
global yval14
global yval15
global parmlo
global parmhi
global parmcur
global linecolor
global graphlist

set gname [index $graphlist $gnum]
if {![winfo exists $gname]} {return}

if {$pnum != "all"} {
    set param($pnum) [string trim $param($pnum)]
    puts "param($pnum) => $param($pnum)"
}

# Update parameter list
set DataSrcList ""
for {set ctr 0} {$ctr < 16} {incr ctr} {
    set value [css_cmd ctrl "getval $param($ctr)"]
    if {$value == "-----"} {set param($ctr) ""}
    if {$param($ctr) == ""} {
        lappend DataSrcList "a_z"
    } {
        lappend DataSrcList $param($ctr)
    }
}
css_cmd stat "setlist $datalistid $DataSrcList"

# Update plot data list for new parameter
set vlist [css_cmd stat "getlist $datalistid"]
if {$pnum == "all"} {
    set p0 [expr $gnum * 4]
    set p1 [expr $p0 + 1]
    set p2 [expr $p0 + 2]
    set p3 [expr $p0 + 3]
    catch {unset xval$p0}
}

```

```

catch {unset xval$P1}
catch {unset xval$P2}
catch {unset xval$P3}
catch {unset yval$P0}
catch {unset yval$P1}
catch {unset yval$P2}
catch {unset yval$P3}
vector xval$P0 xval$P1 xval$P2 xval$P3
vector yval$P0 yval$P1 yval$P2 yval$P3
$gname element configure line$P0 -xdata xval$P0 -ydata yval$P0 -label $param($P0) -color $linecolor(0)
$gname element configure line$P1 -xdata xval$P1 -ydata yval$P1 -label $param($P1) -color $linecolor(1)
$gname element configure line$P2 -xdata xval$P2 -ydata yval$P2 -label $param($P2) -color $linecolor(2)
$gname element configure line$P3 -xdata xval$P3 -ydata yval$P3 -label $param($P3) -color $linecolor(3)
for {set ctr 0; set ctr2 $P0} {ctr < 4} {incr ctr; incr ctr2} {
    set parmcur($ctr2) [lindex $vlist $ctr2]
    set parmlo($ctr2) $parmcur($ctr2)
    set parmhi($ctr2) $parmcur($ctr2)
}
} {
    set newparm $param($Pnum)
    unset xval$Pnum
    unset yval$Pnum
    vector xval$Pnum yval$Pnum
    set colomo [expr $Pnum % 4]
    $gname element configure line$Pnum -data {} -xdata xval$Pnum -ydata yval$Pnum -label $param($Pnum) -color $linecolor($colomo)
    set parmcur($Pnum) [lindex $vlist $Pnum]
    set parmlo($Pnum) $parmcur($Pnum)
    set parmhi($Pnum) $parmcur($Pnum)
}
}

```

Procedure: setranges

```
proc setranges { gnum } {
```

```
global dispmode
```

```
global yscalemode
```

```
global yscalemin
```

```
global yscalemax
```

```
global rangestart
```

```
global rangestop
```

```
global tailsze
```

```
global simsecs
```

```
global minval
```

```
global maxval
```

```
global graphlist
```

```
set gname [lindex $graphlist $gnum]
```

```
if {![winfo exists $gname]} {return}
```

```
switch $dispmode($gnum) {
```

```
all
```

```
{
    $gname xaxis configure -min 0 -max ""
}
```

```
tail
```

```
{
    set minval [expr $simsecs - $tailsze($gnum) / 2]
    if {$minval < 0} {
        set minval 0
    }
    set maxval($gnum) [expr $minval + $tailsze($gnum)]
    $gname xaxis configure -min $minval -max $maxval($gnum)
}
```

```
range
```

```
{
    if {$rangestart >= 0 && $rangestart < $rangestop} {
        $gname xaxis configure -min $rangestart -max $rangestop
    }
}
```

```
switch $yscalemode($gnum) {
```

```
auto
```

```
{
    $gname yaxis configure -min "" -max ""
}
```

```
range
```

```
{
    if {$yscalemin < $yscalemax} {
        $gname yaxis configure -min $yscalemin -max $yscalemax
    }
}
```

```

}
}

# Procedure: setrate
proc setrate {} {
  global freq
  global freq_new
  global cyclicrate
  global interval
  global baserate

  global freq
  global freq_new

  switch $freq_new {
    1 -
    2 -
    3 -
    4 -
    5 {
      set freq $freq_new
      set interval [expr 1000 / $freq]
    }
    userspec {
      if {($interval >= 100) && ($interval <= 2000)} {
        set freq [expr int(1000.0 / $interval)]
      }
    }
    algorithm {
      set freq 4
      set interval 250
      after 6000 {set freq 2; set interval 500}
      after 12000 {set freq 1; set interval 1000; set freq_new 1}
    }
  }
}
}

```

```

# Procedure: showgraphs
proc showgraphs { totgraphs } {
  # This procedure sets the total number of graphs displayed
  global graphlist

  # check for a valid number
  set tnum $totgraphs
  if {[catch {incr tnum -1} result]} {set totgraphs 3}
  if {$totgraphs < 1} {set totgraphs 1}

  # use the packer to control which graphs are displayed
  switch $totgraphs {
    1 {
      pack [lindex $graphlist 0] -side top -expand 1 -fill both
      pack forget [lindex $graphlist 1]
      pack forget .gbase1
    }
    2 {
      pack [lindex $graphlist 0] [lindex $graphlist 1] -side top -expand 1 -fill both
      pack forget .gbase1
    }
    3 {
      pack [lindex $graphlist 0] [lindex $graphlist 1] -side top -expand 1 -fill both
      pack .gbase1 -after .gbase0.graph1 -side top -expand 1 -fill both
      pack [lindex $graphlist 2] -side top -expand 1 -fill both
      pack forget [lindex $graphlist 3]
    }
    4 {
      pack [lindex $graphlist 0] [lindex $graphlist 1] -side left -expand 1 -fill both
      pack forget .gbase1
      pack .gbase1 -side top -expand 1 -fill both
      pack [lindex $graphlist 2] [lindex $graphlist 3] -side left -expand 1 -fill both
    }
  }
}
}

```

```

set gname [lindex $graphlist 0]
if {$!olgraphs == 1} {
    pack forget $gname.frame18
    $gname legend configure -mapped true
    pack .gbase0.frame0 -side top -anchor w
}
{
    pack forget .gbase0.frame0
    pack $gname.frame18 -side bottom -anchor w
    [[index $graphlist 0] legend configure -mapped false
}
}
}

```

User defined images

```

# application parsing procedure
proc XFLocalParseAppDefs {xfAppDefFile} {
    global xfAppDefaults tcl_platform

    # basically from: Michael Moore
    if {[file exists $xfAppDefFile] &&
        [file readable $xfAppDefFile] &&
        "[file type $xfAppDefFile]" == "link"} {
        catch "file type $xfAppDefFile" xfType
        while {"$xfType" == "link"} {
            if {[catch "file readlink $xfAppDefFile" xfAppDefFile]} {
                return
            }
            catch "file type $xfAppDefFile" xfType
        }
    }
    if {!(("$xfAppDefFile" != "" &&
        [file exists $xfAppDefFile] &&
        [file readable $xfAppDefFile] &&
        "[file type $xfAppDefFile]" == "file"))} {
        return
    }
    if {[catch "open $xfAppDefFile r" xfResult]} {
        set xfAppFileContents [read $xfResult]
        close $xfResult
        foreach line [split $xfAppFileContents "\n"] {
            # backup indicates how far to backup. It applies to the
            # situation where a resource name ends in . and when it
            # ends in ". In the second case you want to keep the "
            # in the widget name for pattern matching, but you want
            # to get rid of the . if it is the end of the name.
            set backup -2
            set line [string trim $line]
            if {[string index $line 0] == "#" || "$line" == ""} {
                # skip comments and empty lines
                continue
            }
            if {[string compare "windows" $tcl_platform(platform)]} {
                set list [split $line ";"]
            }
            {
                set list [split $line ".;"]
            }
            set resource [string trim [lindex $list 0]]
            set i [string last "." $resource]
            set j [string last "" $resource]
            if {$j > $i} {
                set i $j
                set backup -1
            }
            incr i
            set name [string range $resource $i end]
            incr i $backup
            set widname [string range $resource 0 $i]
            set value [string trim [lindex $list 1]]
            if {"$widname" != "" && "$widname" != ""} {
                # insert the widget and resourcename to the application
                # defaults list.
                if {[info exists xfAppDefaults]} {
                    set xfAppDefaults ""
                }
            }
        }
    }
}

```



```

        setranges $ctr
    }

    # activate cyclic procedure
    set cyclic_on 1
    cyclic_graph
}

# prepare auto loading
global auto_path
global tk_library
global xflLoadPath
foreach xflElement [eval list [split $xflLoadPath @] $auto_path] {
    if {[file exists $xflElement/tclIndex]} {
        lappend auto_path $xflElement
    }
}

# startup source
StartupSrc

# initialize global variables
proc InitGlobals {} {
    global {DataSrcList}
    set {DataSrcList} {edibattery edivoltreg edravgcell edkrhtr3 a_z a_z a_z a_z a_z a_z a_z a_z a_z a_z a_z}
    global {DataSrcList2}
    set {DataSrcList2} {}
    global {DataValList2}
    set {DataValList2} {}
    global {baserate}
    set {baserate} {1000}
    global {bl_library}
    set {bl_library} {/tcl/lib/bl_library}
    global {bl_versions}
    set {bl_versions(BLT)} {2.1}
    set {bl_versions(barchart)} {4.0}
    set {bl_versions(bitmap)} {2.0}
    set {bl_versions(bldebug)} {0.9}
    set {bl_versions(drag&drop)} {2.1}
    set {bl_versions(graph)} {4.0}
    set {bl_versions(htext)} {3.0}
    set {bl_versions(spline)} {1.0}
    set {bl_versions(table)} {2.0}
    set {bl_versions(tilebutton)} {0.9}
    set {bl_versions(tilecheckboxbutton)} {0.9}
    set {bl_versions(tileframe)} {0.9}
    set {bl_versions(tilelabel)} {0.9}
    set {bl_versions(tileradiobutton)} {0.9}
    set {bl_versions(tiles scrollbar)} {0.9}
    set {bl_versions(tiletoplevel)} {0.9}
    set {bl_versions(vector)} {1.0}
    set {bl_versions(watch)} {1.0}
    set {bl_versions(winop)} {1.1}
    global {clockstr}
    set {clockstr} {14:02:07}
    global {colmsecs}
    set {colmsecs} {1000}
    global {connection}
    set {connection(cyrix)} {}
    set {connection(pentium)} {}
    set {connection(zenon)} {sock3}
    global {cpusage}
    set {cpusage} {0.200000}
    global {ctr}
    set {ctr} {4}
    global {curIC}
    set {curIC} {IC50}
    global {curparm}
    set {curparm} {3}
    global {cyclic_on}
    set {cyclic_on} {1}
    global {cyclicrate}
    set {cyclicrate} {100}
    global {datalistid}
    set {datalistid} {37}
}

```

```

global {datalistid2}
set {datalistid2} {38}
global {datasrc}
set {datasrc} {sock3}
global {datestr}
set {datestr} {08/28/97}
global {display}
set {display} {tail}
global {dispmode}
set {dispmode(0)} {all}
set {dispmode(1)} {all}
set {dispmode(2)} {all}
set {dispmode(3)} {all}
global {disprange}
set {disprange} {tail}
global {fin}
set {fin} {file8}
global {fou1}
set {fou1} {file8}
global {freq}
set {freq} {2}
global {freq_new}
set {freq_new} {1}
global {gname}
set {gname} {gbase3.frame8.frame18}
global {gnum}
set {gnum} {0}
global {graphlist}
set {graphlist} {gbase0.graph0 .gbase0.graph1 .gbase1.graph0 .gbase1.graph2}
global {index}
set {index} {7}
global {interval}
set {interval} {1000}
global {item}
set {item} {2095}
global {line}
set {line} {# Graph parameters}
global {linecolor}
set {linecolor(0)} {black}
set {linecolor(1)} {blue}
set {linecolor(2)} {red}
set {linecolor(3)} {orange}
global {list_id}
set {list_id(cyrix)} {}
set {list_id(pentium)} {}
set {list_id(zenon)} {37}
global {list_id2}
set {list_id2(cyrix)} {8}
set {list_id2(pentium)} {}
set {list_id2(zenon)} {38}
global {load_name}
set {load_name} {}
global {machine}
set {machine} {zenon}
global {maxval}
set {maxval(0)} {60}
set {maxval(1)} {60}
set {maxval(2)} {60}
set {maxval(3)} {60}
global {minval}
set {minval} {0}
global {nlist}
set {nlist} {xtemp1 xtemp2}
global {p0}
set {p0} {4}
global {param}
set {param(0)} {edibattery}
set {param(1)} {edivoltreg}
set {param(10)} {}
set {param(11)} {}
set {param(12)} {}
set {param(13)} {}
set {param(14)} {}
set {param(15)} {}
set {param(2)} {edravgcell}

```

```

set {param(3)} {edkhr3}
set {param(4)} {}
set {param(5)} {}
set {param(6)} {}
set {param(7)} {}
set {param(8)} {}
set {param(9)} {}
global {param1}
set {param1} {one}
global {parmcur}
set {parmcur(0)} {1.1732}
set {parmcur(1)} {17}
set {parmcur(10)} {-----}
set {parmcur(11)} {-----}
set {parmcur(12)} {-----}
set {parmcur(13)} {-----}
set {parmcur(14)} {-----}
set {parmcur(15)} {-----}
set {parmcur(2)} {0.008}
set {parmcur(3)} {10}
set {parmcur(4)} {-----}
set {parmcur(5)} {-----}
set {parmcur(6)} {-----}
set {parmcur(7)} {-----}
set {parmcur(8)} {-----}
set {parmcur(9)} {-----}
global {parmdisp}
set {parmdisp(0)} {cur}
set {parmdisp(1)} {cur}
set {parmdisp(10)} {cur}
set {parmdisp(11)} {cur}
set {parmdisp(12)} {cur}
set {parmdisp(13)} {cur}
set {parmdisp(14)} {cur}
set {parmdisp(15)} {cur}
set {parmdisp(2)} {cur}
set {parmdisp(3)} {cur}
set {parmdisp(4)} {cur}
set {parmdisp(5)} {cur}
set {parmdisp(6)} {cur}
set {parmdisp(7)} {cur}
set {parmdisp(8)} {cur}
set {parmdisp(9)} {cur}
global {parmhi}
set {parmhi(0)} {2.2405}
set {parmhi(1)} {17}
set {parmhi(10)} {-----}
set {parmhi(11)} {-----}
set {parmhi(12)} {-----}
set {parmhi(13)} {-----}
set {parmhi(14)} {-----}
set {parmhi(15)} {-----}
set {parmhi(2)} {0.008}
set {parmhi(3)} {10}
set {parmhi(4)} {-----}
set {parmhi(5)} {-----}
set {parmhi(6)} {-----}
set {parmhi(7)} {-----}
set {parmhi(8)} {-----}
set {parmhi(9)} {-----}
global {parmlo}
set {parmlo(0)} {1.1732}
set {parmlo(1)} {17}
set {parmlo(10)} {-----}
set {parmlo(11)} {-----}
set {parmlo(12)} {-----}
set {parmlo(13)} {-----}
set {parmlo(14)} {-----}
set {parmlo(15)} {-----}
set {parmlo(2)} {0.008}
set {parmlo(3)} {10}
set {parmlo(4)} {-----}
set {parmlo(5)} {-----}
set {parmlo(6)} {-----}
set {parmlo(7)} {-----}

```

```
set {pamlo(8)} {-----}
set {pamlo(9)} {-----}
global {pamval}
set {pamval(0)} {1.1732}
set {pamval(1)} {17}
set {pamval(10)} {}
set {pamval(11)} {}
set {pamval(12)} {}
set {pamval(13)} {}
set {pamval(14)} {}
set {pamval(15)} {}
set {pamval(2)} {}
set {pamval(3)} {}
set {pamval(4)} {}
set {pamval(5)} {}
set {pamval(6)} {}
set {pamval(7)} {}
set {pamval(8)} {}
set {pamval(9)} {}
global {plotbuf_x}
set {plotbuf_x(0)} {2094 2095}
set {plotbuf_x(1)} {}
set {plotbuf_x(2)} {}
set {plotbuf_x(3)} {}
set {plotbuf_x(4)} {2094 2095}
set {plotbuf_x(8)} {2094 2095}
global {plotbuf_y}
set {plotbuf_y(0)} {-1.5368 -1.5277}
set {plotbuf_y(1)} {}
set {plotbuf_y(2)} {}
set {plotbuf_y(3)} {}
set {plotbuf_y(4)} {1.9798 1.9799}
set {plotbuf_y(8)} {0.008 0.008}
global {plotbutton}
set {plotbutton} {Suspend Plotting}
global {plotting}
set {plotting} {1}
global {port}
set {port} {5002}
global {query_match}
set {query_match} {e}
global {query_type}
set {query_type} {start}
global {rangestart}
set {rangestart} {1000}
global {rangestop}
set {rangestop} {1200}
global {result}
set {result} {}
global {sampletype}
set {sampletype} {userspec}
global {simrate}
set {simrate} {5.00}
global {simsecs}
set {simsecs} {470}
global {simsecslast}
set {simsecslast} {470}
global {simtime}
set {simtime} {00:07:50}
global {standalone}
set {standalone} {0}
global {status}
set {status} {Run}
global {tailsizes}
set {tailsizes(0)} {60}
set {tailsizes(1)} {60}
set {tailsizes(2)} {60}
set {tailsizes(3)} {60}
global {tailsizetmp}
set {tailsizetmp} {60}
global {totalgraphs}
set {totalgraphs} {1}
global {xval0}
set {xval0(++end)} {470}
set {xval0(end)} {470.0}
```

```

global {xval1}
set {xval1(++end)} {470}
set {xval1(end)} {470.0}
global {xval10}
set {xval10(end)} {0.0}
global {xval11}
set {xval11(end)} {0.0}
global {xval12}
set {xval12(end)} {0.0}
global {xval13}
set {xval13(end)} {0.0}
global {xval14}
set {xval14(end)} {0.0}
global {xval15}
set {xval15(end)} {0.0}
global {xval2}
set {xval2(end)} {0.0}
global {xval3}
set {xval3(end)} {0.0}
global {xval4}
set {xval4(end)} {0.0}
global {xval5}
set {xval5(end)} {0.0}
global {xval6}
set {xval6(end)} {0.0}
global {xval7}
set {xval7(end)} {0.0}
global {xval8}
set {xval8(end)} {0.0}
global {xval9}
set {xval9(end)} {0.0}
global {yscale}
set {yscale} {auto}
global {yscalemax}
set {yscalemax} {30}
global {yscalemin}
set {yscalemin} {0}
global {yscalemode}
set {yscalemode(0)} {auto}
set {yscalemode(1)} {auto}
set {yscalemode(2)} {auto}
set {yscalemode(3)} {auto}
global {ystart}
set {ystart} {100}
global {yval0}
set {yval0(++end)} {1.1732}
set {yval0(end)} {1.1732}
global {yval1}
set {yval1(++end)} {17}
set {yval1(end)} {17.0}
global {yval10}
set {yval10(end)} {0.0}
global {yval11}
set {yval11(end)} {0.0}
global {yval12}
set {yval12(end)} {0.0}
global {yval13}
set {yval13(end)} {0.0}
global {yval14}
set {yval14(end)} {0.0}
global {yval15}
set {yval15(end)} {0.0}
global {yval2}
set {yval2(end)} {0.0}
global {yval3}
set {yval3(end)} {0.0}
global {yval4}
set {yval4(end)} {0.0}
global {yval5}
set {yval5(end)} {0.0}
global {yval6}
set {yval6(end)} {0.0}
global {yval7}
set {yval7(end)} {0.0}
global {yval8}

```

```

set {yval8(end)} {0.0}
global {yval9}
set {yval9(end)} {0.0}

# please don't modify the following
# variables. They are needed by xf.
global {autoLoadList}
set {autoLoadList(icsgraph.tcl)} {}
set {autoLoadList(main.tcl)} {}
global {internalAliasList}
set {internalAliasList} {}
global {moduleList}
set {moduleList(icsgraph.tcl)} {}
global {preloadList}
set {preloadList(xfiInternal)} {}
global {symbolicName}
set {symbolicName(root)} {}
global {xfWmSetPosition}
set {xfWmSetPosition} {,samplerate ,range}
global {xfWmSetSize}
set {xfWmSetSize} {,samplerate ,range .}
global {xfAppDefToplevels}
set {xfAppDefToplevels} {}
}

# initialize global variables
InitGlobals

# display/remove toplevel windows.
ShowWindow.

global xfShowWindow.query
set xfShowWindow.query 0

global xfShowWindow.range
set xfShowWindow.range 0

global xfShowWindow.samplerate
set xfShowWindow.samplerate 0

# load default bindings.
if {[info exists env(XF_BIND_FILE)] &&
    "[info procs XFShowHelp]" == ""} {
    source $env(XF_BIND_FILE)
}

# initialize bindings for all widgets.
XFInitAllBindings

# parse and apply application defaults.
XFLocalLoadAppDefs icsgraph
XFLocalSetAppDefs

# end source
EndSrc

# eof
#

# -----
# Copyright (C) 1996 Dennis R. LaBelle (drlabel@albany.net) All Rights Reserved.
#
# PERMISSION IS GRANTED TO DISTRIBUTE THIS SOFTWARE FREELY, WITH THE
# EXCEPTION THAT ONE MAY NOT CHARGE FOR IT OR INCLUDE IT WITH SOFTWARE
# WHICH IS SOLD. Permission to use, copy, modify, and distribute this software and
# its documentation for educational, research, internal corporate and non-profit
# purposes, without fee, and without a written agreement is hereby granted for all
# cases that do not conflict with the restriction in the first sentence of this
# paragraph, provided that the above copyright notice, this paragraph, and the
# following three paragraphs appear in all copies.
#
# Permission to incorporate this software into commercial products may be obtained
# from the author (e-mail drlabel@albany.net).
#

```

```
# IN NO EVENT SHALL THE AUTHOR BE LIABLE TO ANY PARTY FOR DIRECT,  
# INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF  
# THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE AUTHOR HAS  
# BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
#
```

```
# THE AUTHOR SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT  
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A  
# PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS"  
# BASIS, AND THE AUTHOR HAS NO OBLIGATION TO PROVIDE MAINTENANCE,  
# SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
```

```
#
```

```
# Procedure: cyclic
```

```
proc cyclic {} {# xf ignore me 9
```

```
global cyclicrate
```

```
global cyclic_on
```

```
if {$cyclic_on} {
```

```
    if {[info exists cyclicrate]} {set cyclicrate 1000}
```

```
    cyclic_user
```

```
    after $cyclicrate cyclic
```

```
}
```

```
}
```

```
# Procedure: query_list
```

```
proc query_list { matchstr } {# xf ignore me 9
```

```
global query_type
```

```
set qlist [css_cmd stat "find$query_type $matchstr"]
```

```
.query.frame.listbox1 delete 0 end
```

```
foreach item $qlist {
```

```
    .query.frame.listbox1 insert end $item
```

```
}
```

```
}
```

```
proc quick_cyclic {} {# xf ignore me 9
```

```
global cyclicrate
```

```
global cyclic_on
```

```
.if {$cyclic_on} {
```

```
    if {[info exists cyclicrate]} {set cyclicrate 1000}
```

```
    getval_cyclic
```

```
    cyclic_user
```

```
    after $cyclicrate quick_cyclic
```

```
}
```

```
}
```

```
# Procedure: css_cmd
```

```
proc css_cmd {cmdtype cmdstr} {# xf ignore me 9
```

```
global datasrc
```

```
if {$datasrc == ""} {return ""}
```

```
puts $datasrc "$cmdtype $cmdstr"
```

```
flush $datasrc
```

```
gets $datasrc result
```

```
return $result
```

```
}
```

```
proc css_stat {cmd} {
```

```
css_cmd stat $cmd
```

```
}
```

```
# Procedure: buildlist
```

```
proc buildlist { wname } {# xf ignore me 9
```

```
global DataSrcList2
```

```
global datalistid2
```

```
if {[winfo exists $wname]} {
```

```
    set DataSrcList2 [findvars $wname]
```

```
} {
```

```
    set DataSrcList2 ""
```

```
}
```



```

css_cmd stat "setlist $datalistid2 $DataSrcList2"
}

proc quick_connect {} {# xf ignore me 9
global connection
global datasrc
global DataSrcList
global DataSrcList2
global list_id
global list_id2
global cyclic_on
global cyclicrate
global port
global INVALID
global argv
global argv0

# establish clean up and quit function
wm protocol . WM_DELETE_WINDOW {quick_quit}

# Connect to local machine if host not specified
if {$argv == ""} {
    set host [info hostname]
} {set host [lindex $argv 0]}

# set up load host and TCP/IP socket port number
if {$argv != ""} {set load_host [lindex $argv 0]}
set port [lindex $argv 1]
if {$port == ""} {set port 5000}

### check for valid port number
set temp $port
if {[catch {incr temp} result]} {set port 5000}
if {$port < 5000 || $port > 5999} {set port 5000}

# clear out any old connections
set namelist [array names connection]
foreach item $namelist {
    set connection($item) ""
    set list_id($item) ""
    set list_id2($item) ""
}

# connect to specified server host
set DataSrcList ""
set DataSrcList2 ""
set datasrc ""
if {[file tail $argv0] != "xfmain.tcl"} {
    switchdest $host
}

# start cyclic retrieval of data
if {$datasrc != ""} {
    buildlist .
    set cyclicrate 1000
    set cyclic_on 1
    set INVALID "____"
    quick_cyclic
}
}

proc feedback {cmd} {# xf ignore me 9
# Execute a command fed back through the data link
#
switch -- $cmd {
    quit {
        quick_quit
    }
    popup {
        wm withdraw .
        wm deiconify .
    }
}
}

```

```

# Procedure: getval_cyclic
proc getval_cyclic {} {# xf ignore me 9
global DataSrcList2
global DataValList2
global datalistid2
global INVALID

set DataValList2 [css_cmd stat "gellist $datalistid2"]
set ctr 0
foreach vname $DataSrcList2 {
    set value [lindex $DataValList2 $ctr]
    if {$value != $INVALID} {
        set cmd "set $vname $value"
        uplevel #0 $cmd
    }
    incr ctr
}
set cmd [lrange $DataValList2 $ctr end]
if {$cmd != ""} {
    feedback $cmd
}
}

```

```

# Procedure: findvars
proc findvars { wname } {# xf ignore me 9
set rtnval ""
set wlist [winfo children $wname]
foreach item $wlist {
    if {[string range $item 0 2] != ".xp"} {
        set result [findvars $item]
        if {$result != ""} {set rtnval [concat $rtnval $result]}
    }
}
if {[winfo class $wname] == "Label"} {
    set tname [$wname cget -textvariable]
    if {$tname != ""} {
        set rtnval [concat $rtnval $tname]
    }
} elseif {[winfo class $wname] == "Scale"} {
    set tname [$wname cget -variable]
    if {$tname != ""} {
        set rtnval [concat $rtnval $tname]
    }
}
return $rtnval
}

```

```

proc quick_quit {} {# xf ignore me 9
global connection
global list_id
global list_id2
global cyclic_on
global argv0

set cyclic_on 0
set conlist [array names connection]
foreach item $conlist {
    if {$connection($item) != ""} {
        css_cmd stat "newlist $list_id($item)"
        css_cmd stat "newlist $list_id2($item)"
        css_cmd stat "disconnect [info hostname]:[file tail $argv0]"
        close $connection($item)
    }
}
exit
}

```

```

# Procedure: ics_quit
proc ics_quit {} {# xf ignore me 9
global connection
global list_id
global list_id2

```

```

global cyclic_on
global argv0

set answer [tk_messageBox -message "Really quit [file tail $argv0]?" -type yesno -icon question]
switch $answer {
    yes {
        set cyclic_on 0
        set conlist [array names connection]
        foreach item $conlist {
            if {$connection($item) != ""} {
                catch {css_cmd stat "newlist $list_id($item)}
                catch {css_cmd stat "newlist $list_id2($item)}
                catch {css_cmd stat "disconnect [info hostname]:[file tail $argv0]}
                catch {close $connection($item)}
            }
        }
        exit
    }
    no {}
}
}

```

```

# Procedure: connect_to_server
proc connect_to_server { machine port } {# xf ignore me 9

    # check whether target machine is on network
    set response [eval "exec ping -n 1 $machine"]

    if {[lsearch -exact $response Reply] == -1} {
        return ""
    } { # try to connect to server
        if {[catch {socket $machine $port} sockchannel]} {
            set sockchannel ""
        }
    }
    return $sockchannel
}

```

```

# Procedure: switchdest
proc switchdest { host } {# xf ignore me 9
global port
global datasrc
global datalistid
global datalistid2
global connection
global list_id
global list_id2
global machine
global DataSrcList
global DataSrcList2
global argv0

if {[info exists connection($host)]} {
    set connection($host) ""
}
if {[info exists list_id($host)]} {
    set list_id($host) ""
    set list_id2($host) ""
}
if {$connection($host) == ""} {
    set connection($host) [connect_to_server $host $port]
    if {$connection($host) != ""} {
        fconfigure $connection($host) -blocking 1 -buffering full
        set datasrc $connection($host)
        set machine $host

        # create new data list on server
        set fname [file tail $argv0]
        set list_id($host) [css_cmd stat "newlist"]
        set list_id2($host) [css_cmd stat "newlist"]
        css_cmd stat "register [info hostname]:[file tail $argv0] $list_id2($host)"
        set datalistid $list_id($host)
        set datalistid2 $list_id2($host)
    }
}

```

```
# send variable list to server
css_cmd stat "setlist $datalistid $DataSrcList"
css_cmd stat "setlist $datalistid2 $DataSrcList2"
}
}
set datasrc $connection($host)
set machine $host
set datalistid $list_id($host)
set datalistid2 $list_id2($host)
}
}
```

```
proc get_odspath {} {# xf ignore me 9
global env
```

```
if {[info exists env(ODSPATH)]} {
    regsub -all {\} $env(ODSPATH) / odspath
} {set odspath [pwd]}
return $odspath
}
```