FiveLinux

The GUI library for Harbour / xHarbour on Linux

www.fivetechsoft.com

CONTENTS

What is FiveLinux for Harbour ?

FiveLinux for Harbour (**FLH**) is a set of commands, classes and functions developed with the objective to obtain the highest xbase development power from the Linux GUI environment (based on **GTK+** www.gtk.org) in conjunction with Harbour, the open source Clipper language compatible compiler (http://www.harbour-project.org).

FLH is a library that you may link to your Harbour built application enabling the use of all these functionalities. FLH has been developed using Harbour and gcc (the Linux C/C++ language compiler gcc.gnu.org). gcc is usually included on all Linux distributions. FiveLinux is also available (**FLX**) for xHarbour (www.xharbour.org) being 100% source code compatible with FLH. You may decide which one you prefer to use. They work exactly the same.

FLH has been developed to achieve the highest compatibility level with the original FiveWin library for CA-Clipper and FiveWin for Harbour. With this first FiveLinux version you should not expect to literally port an entire FiveWin application, as it is, into FiveLinux but you can use it to build applications using the typical familiar FiveWin syntax.

The classes of FLH are completely self contained. FLH provides the needed classes to create the typical GUI windows, dialogboxes and controls, as well as providing a class to manage the printer. FiveTech decided to use GTK+ as the GUI to build FiveLinux on top of it for several reasons:

- GTK+ has been used to develop the Gnome GUI (www.gnome.org).

- The GTK+ license (LGPL) permits the use of it to develop proprietary software without paying any license fees or royalties.
- GTK+ is available on practically all Linux distributions.
- There are lots of technical references about GTK+ and samples on the Internet, as well as a tech support IRC forum at irc.gnome.org #gtk+ channel.
- Glade, the free user interface builder for GTK+ (glade.gnome.org) allows the visual design of dialogboxes with controls, to be used with FiveLinux.

Simple and easy to use: The FiveLinux commands and classes have been specially designed to offer a great ease of use and require a minimum learning curve. FiveWin has always been very popular for its power and ease of use.

These reasons show the enormous importance of using FiveLinux to port your existing Clipper/Harbour and FiveWin apps to Linux. FLH gives you total control over the Linux GUI environment, a must for any professional programming company, letting you deliver professional Linux applications that offer modern and updated management solutions.

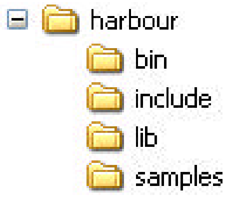
Installing FiveLinux for Harbour

FiveLinux is designed to be used with Harbour (or xHarbour), the open source Clipper language compatible compiler available from:

www.harbour-project.org.

Though you may download the latest build from that site, we strongly recommend you use the one we provide as it has been tested and is already built containing its executable files and libraries ready to be used.

Just click on the provided Harbour installer (install.exe) and it will be automatically installed at /home/<user>/harbour with the following subdirectory structure:

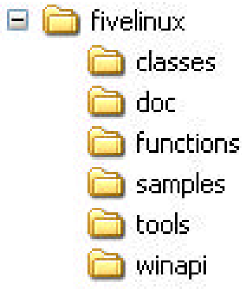


```
harbour
    bin
    include
    lib
    samples
```

Though Linux users recommend installing the binaries at usr/share/bin and the libraries at usr/share/lib, etc. FiveTech recommends the classic Clipper directory structure, as shown in the image, to manage Harbour in an easier way.

We also provide an xHarbour installer (installx.exe) which will automatically install to /home/<user>/xharbour.

Once you have installed Harbour, you have to install FiveLinux. We provide a fivelinux.exe installation application that will install FiveLinux at /home/<user>/fivelinux.



```
fivelinux
    classes
    doc
    functions
    samples
    tools
    winapi
```

As you see it resembles the classic FiveWin directory structure. Both FLH and FLX libraries are installed to select between them simply use samples/build.sh or samples/buildx.sh.

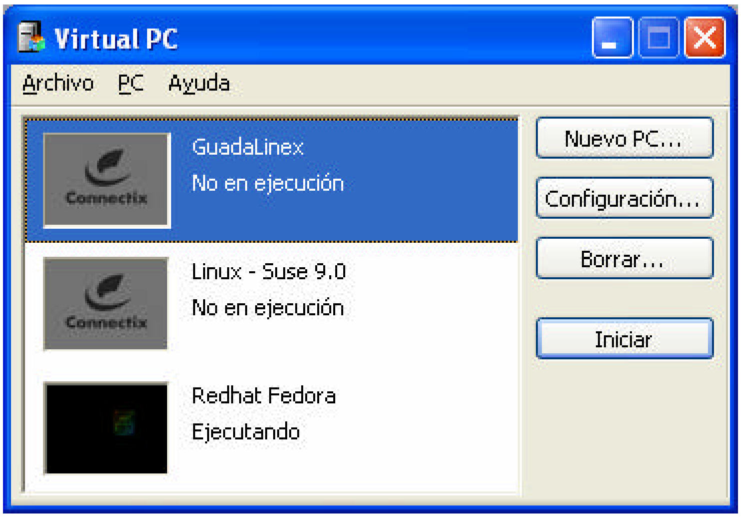
Setting the development environment

Using Windows and Linux together: Microsoft Virtual PC

Many of us are coming to Linux from Windows. We have decided to learn Linux to expand our market possibilities, but we need to keep our existing Windows operating systems and also be able to use Linux.

For such purpose there is an excellent tool that FiveTech highly recommends: Microsoft Virtual PC www.microsoft.com/windowsxp/virtualpc.

Virtual PC is an application that you install on Windows XP Professional (there is a free 45 day evaluation version provided by Microsoft, that we include on our FiveLinux CD). It lets you create Virtual PCs, each one contained on a single file on disk, that don't affect your hard disk at all or your current Windows XP settings. Virtual PC easily guides you to setup a new operating system on a new virtual pc. Once you install it, Virtual PC looks like this:



GuadaLinex
No en ejecución

Linux - Suse 9.0
No en ejecución

Redhat Fedora
Ejecutando

and the entire installed Linux is contained in a window. Pressing Alt-Gr + Enter maximizes it so Windows is no longer visible. To go back to Windows you just press Alt-Gr again. In this manual we explain how to get the best functionality from Virtual PC.

Recommended development tools on Linux

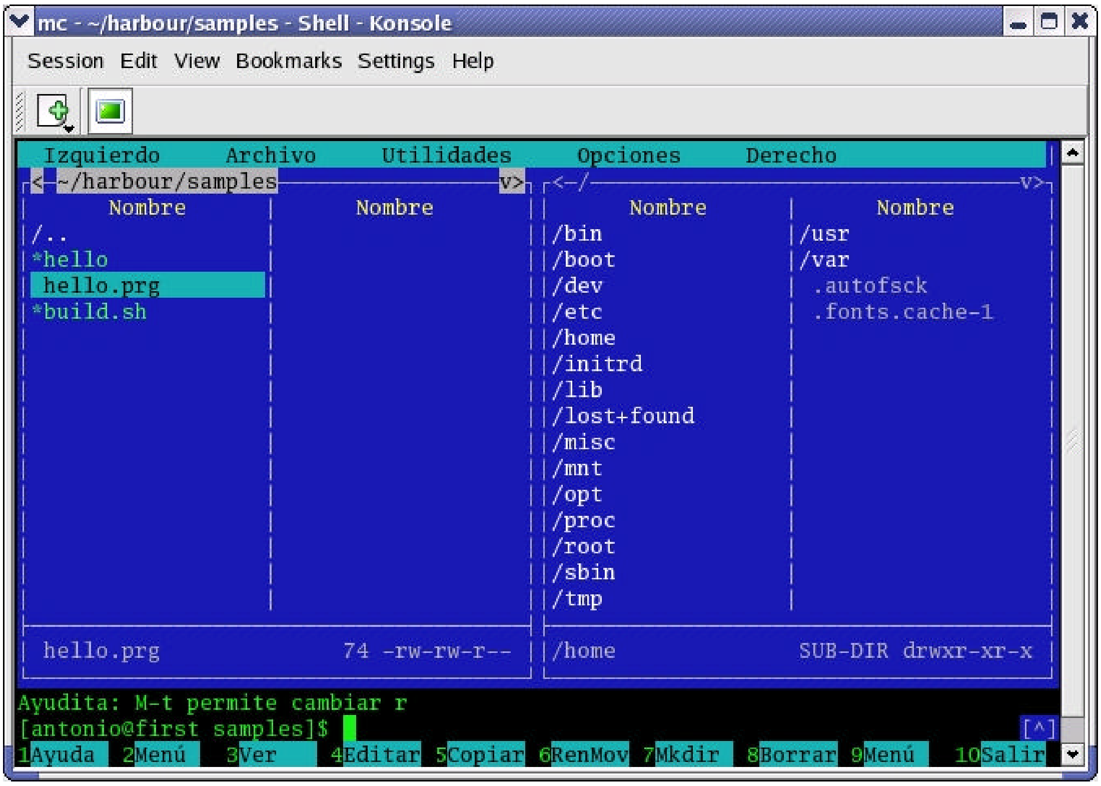
For those of you who are new to Linux, there are several tools that will highly increase your Linux management capabilities:

Midnight Commander:

Midnight Commander is a Linux port of the famous Norton Commander for MsDos. It lets you review the contents of the (virtual, if you use Virtual PC) hard disk, change directories, create them, review files, create them, delete them and much more useful things, without having to know the Linux console mode commands to perform such operations.

Most of the Linux distributions include Midnight Commander: just execute **mc** and you will see if it is provided or not. You may also look for the option “create a console with midnight commander”, which used to be included on Linux menus.

In case it is not included in your Linux distribution, you may download it from www.ibiblio.org/mc. A typical **mc** session:


mc - ~/harbour/samples - Shell - Konsole
Session Edit View Bookmarks Settings Help

```
  Izquierdo     Archivo     Utilidades     Opciones     Derecho
┌<─~/harbour/samples──────────────────v>┐┌<─/─────────────────────────────────v>┐
│      Nombre       │      Nombre       ││      Nombre       │      Nombre       │
│/..                │                   ││/bin               │/usr               │
│*hello             │                   ││/boot              │/var               │
│ hello.prg         │                   ││/dev               │ .autofsck         │
│*build.sh          │                   ││/etc               │ .fonts.cache-1    │
│                   │                   ││/home              │                   │
│                   │                   ││/initrd            │                   │
│                   │                   ││/lib               │                   │
│                   │                   ││/lost+found        │                   │
│                   │                   ││/misc              │                   │
│                   │                   ││/mnt               │                   │
│                   │                   ││/opt               │                   │
│                   │                   ││/proc              │                   │
│                   │                   ││/root              │                   │
│                   │                   ││/sbin              │                   │
│                   │                   ││/tmp               │                   │
├───────────────────────────────────────┤├───────────────────────────────────────┤
│ hello.prg               74 -rw-rw-r-- ││/home                  SUB-DIR drwxr-xr-x │
└───────────────────────────────────────┘└───────────────────────────────────────┘
Ayudita: M-t permite cambiar r
[antonio@first samples]$                                                     [^]
1Ayuda  2Menú   3Ver    4Editar 5Copiar 6RenMov 7Mkdir  8Borrar 9Menú   10Salir
```

Midnight Commander will use the language according to your Linux language settings.

Another very important functionality of Midnight Commander is that it lets you review a FTP connection on one of its panels. This is a very easy way to access your Windows XP directories if you need to. Simply start a FTP server on your Windows XP (we do recommend the free CesarFTP accessible from www.aclogic.com) and select the FTP link accessible from Midnight Commander. You will be able to read the FTP shared directories, read files, send files, etc.

Kate:

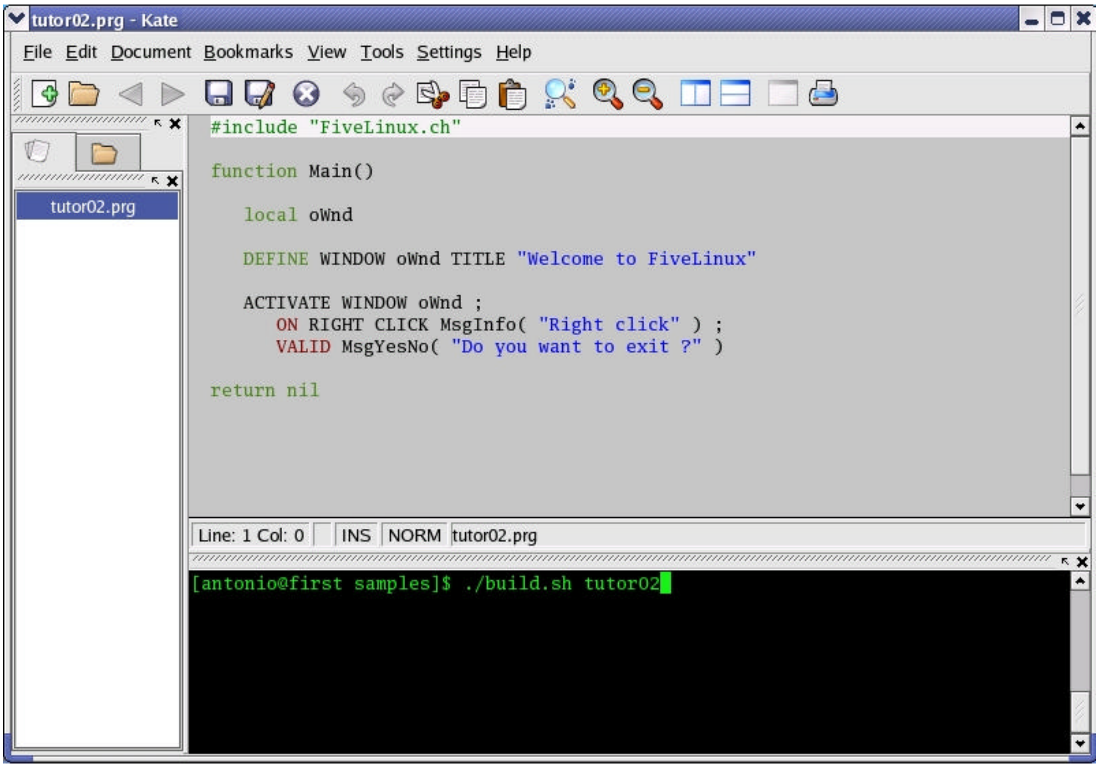
Kate (KDE Advanced Text Editor) is the source code editor we have used to develop FiveLinux. Most Linux distributions include it, but in case yours does not you may do an:

```
apt-get install kate
```

And it will be installed on your Linux. apt-get is a typical utility to download and install new packages and upgrades on Debian based Linux distributions. In case your Linux distribution does not include apt-get, you may download it from:

http://packages.debian.org/stable/base/apt

Kate offers a source code window, a files selection window and a console view, where you can easily test your application. This window structure really speeds up your development cycle for FiveLinux applications. As you can see on the below image, you can build and test your application from the console window of Kate:

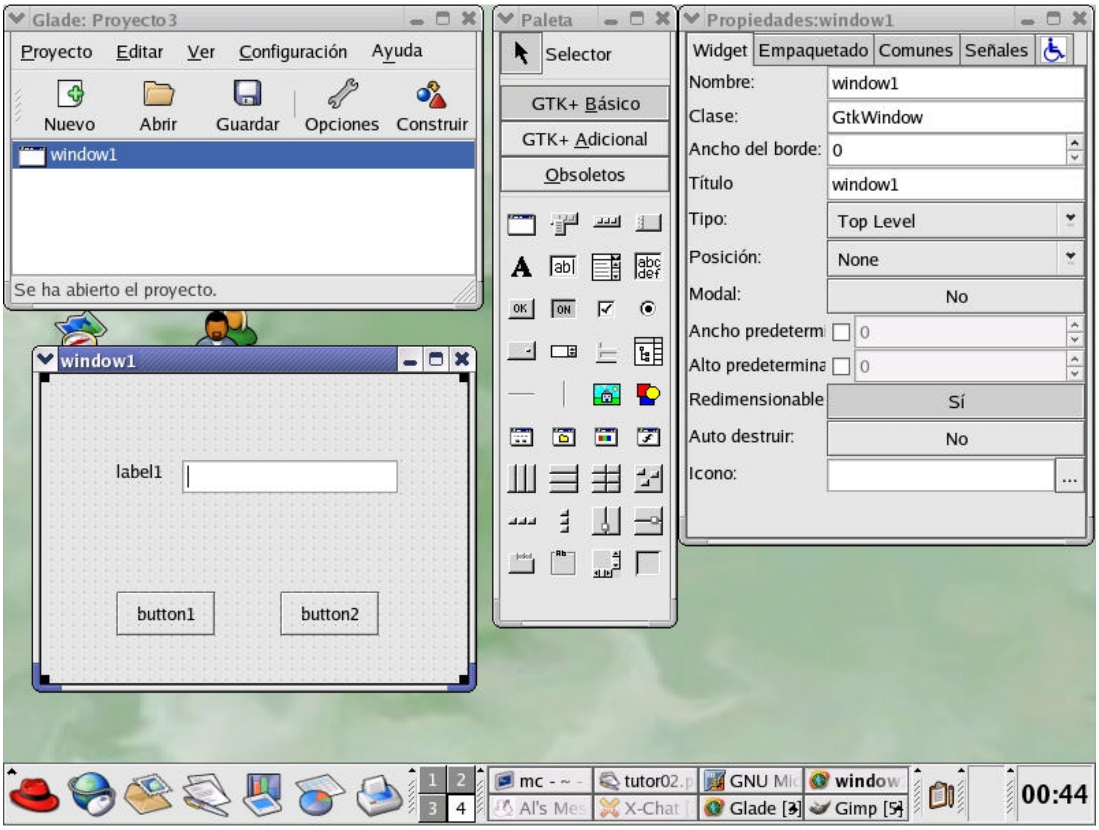

tutor02.prg - Kate
File Edit Document Bookmarks View Tools Settings Help
tutor02.prg
Line: 1 Col: 0 INS NORM tutor02.prg

```
#include "FiveLinux.ch"

function Main()

   local oWnd

   DEFINE WINDOW oWnd TITLE "Welcome to FiveLinux"

   ACTIVATE WINDOW oWnd ;
      ON RIGHT CLICK MsgInfo( "Right click" ) ;
      VALID MsgYesNo( "Do you want to exit ?" )

return nil
```

```
[antonio@first samples]$ ./build.sh tutor02
```

Glade:

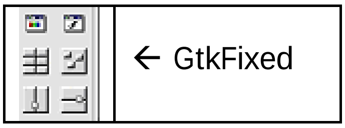
Glade, the free user interface builder for GTK+ (glade.gnome.org) allows the visual design of dialog boxes with controls, to be used with FiveLinux. A typical Glade session:



Se ha abierto el proyecto.

Nombre: window1
Clase: GtkWindow
Ancho del borde: 0
Título window1
Tipo: Top Level
Posición: None
Modal: No
Ancho predeterm □ 0
Alto predetermina □ 0
Redimensionable Sí
Auto destruir: No
Icono: ...

WARNING: To design FiveLinux dialog boxes, you need to place a GtkFixed component as the first element to be placed on the window:



FiveLinux automatically manages this component, so you don't have to worry about it or to do some special coding to manage it. FiveLinux takes care of it.

Packages required to develop with FiveLinux:

FiveLinux applications use the GTK+ libraries that usually are already installed on Linux, but for FiveLinux applications development you must have the GTK+ development libraries too. In order to install them, you may do a:

```
apt-get install gtk2-devel
```

from a shell console window (remember to first login as supervisor typing ‘su’ and you will be prompted for the root login).

You also need to install Gnome-Print for printing:

```
apt-get install libgnomeprintui22-devel
```

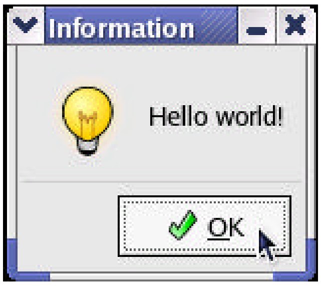
Using FiveLinux for Harbour

Once you have installed Harbour and FiveLinux, simply go to /home/<user>/FiveLinux/samples directory and type:

```
./build.sh tutor01
```

This should call Harbour to compile tutor01.prg and also gcc to generate the required OBJ, and finally build the application. This whole process is performed in an automatic way from build.sh.

A first Hello World! GUI small dialog should appear on the screen:



Hello world!

If not, please review that Harbour is properly located at /home/<user>/harbour and FiveLinux is located at /home/<user>/fivelinux. If you have decided to install them at a different location then you may need to change the paths inside build.sh (ascii file) to ensure they are the right ones.

We do recommend that you continue testing the different samples we provide on the samples directory.

Guidelines for Harbour users

FLH is contained into two libraries: libfive.a and libfivec.a. We also provide a libfivex.a that is just required for xHarbour, as both Harbour and xHarbour use the same libfivec.a library. Please notice that on Linux the libraries have the .a extension.

Besides that, FLH provides an extensive set of header files (.CH extension) that implements all required xBase commands to create and manage all the different GUI objects.

In our samples directory '/samples', we provide the file 'build.sh' for Harbour and buildx.sh for xHarbour. This file shows the correct way to compile and link the applications developed with Harbour and FLH.

build.sh:

```
# ./build.sh

clear

if [ $# = 0 ]; then
   echo syntax: ./build.sh file [options...]
   exit
fi

echo compiling...
./../../harbour/bin/harbour    $1    -n    -I./../include    -
I./../../harbour/include $2
```

```
echo compiling C module...
gcc  $1.c  -c  -I./../include  -I./../../harbour/include  `pkg-
config --cflags gtk+-2.0`

echo linking...
gcc $1.o -o$1 -L./../lib -L./../../harbour/lib `pkg-config --
libs gtk+-2.0` `pkg-config --libs libgnomeprintui-2.2` -lfive
-lfivec -ldebug -lvm -lrtl -lgtcrs -lncurses -llang -lrdd -
lrtl -lvm -lmacro -lpp -ldbfntx -ldbfcdx -ldbfdbt -lcommon -
lgpm -lm

rm $1.c
rm $1.o

echo done!
./$1
```

Please notice the use of gcc after calling the Harbour compiler, to generate the appropriate OBJ file to be linked with the gcc linker feature.

FiveLinux Architecture

FiveLinux provides the same three layer architecture as FiveWin:

- xBase commands: Highest productivity level.
- Classes: Object Oriented management.
- Functions: C wrappers to GTK+, Gnome-Print and to C-RTL functions.

xBase commands:

Window management:

```
DEFINE WINDOW <oWnd> ;
            [ TITLE <cTitle> ] ;
            [ MENU <oMenu> ] ;
            [ SIZE <nWidth>, <nHeight> ]

ACTIVATE WINDOW <oWnd> ;
            [ VALID <uValid> ] ;
            [ ON [ LEFT ] CLICK <uLClick> ] ;
            [ ON RIGHT CLICK <uRClick> ] ;
            [ MAXIMIZED ]
```

Dialog Box management:

```
DEFINE DIALOG <oDlg> ;
            [ TITLE <cTitle> ] ;
            [ SIZE <nWidth>, <nHeight> ]

ACTIVATE DIALOG <oDlg> ;
            [ VALID <uValid> ] ;
            [ ON [ LEFT ] CLICK <uLClick> ] ;
            [ ON RIGHT CLICK <uRClick> ] ;
            [ CENTER | CENTERED ] ;
            [ NOWAIT | NOMODAL ]
```

Control management:

```
DEFINE BUTTONBAR [<oBar>] ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ SIZE <nWidth>, <nHeight> ]

DEFINE BUTTON [<oBtn>] ;
                    [ OF <oBar> ] ;
                    [ LABEL | PROMPT <cText> ] ;
                    [ RESOURCE | NAME | RESNAME <cResName> ] ;
                    [ ACTION <uAction> ] ;
                    [ GROUP ]

@ <nRow>, <nCol> BROWSE <oBrw> ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ HEAD | HEADER | HEADERS | TITLE <cHeading,...> ] ;
                    [ FIELDS <Expr1> [,<ExprN>] ] ;
                    [ ALIAS <cAlias> ] ;
                    [ SIZE <nWidth>, <nHeight> ] ;
                    [ UPDATE ]

@ <nRow>, <nCol> BUTTON [ <oBtn> PROMPT ] <cPrompt> ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ RESOURCE | NAME | RESNAME <cResName> ] ;
```

```
                     [ ACTION <uAction> ] ;
                     [ SIZE <nWidth>, <nHeight> ] ;
                     [ VALID <uValid> ] ;
                     [ WHEN <uWhen> ] ;
                     [ UPDATE ]

@ <nRow>, <nCol> CHECKBOX [ <oCbx> VAR ] <lVar> ;
                     [ OF | WINDOW | DIALOG <oWnd> ] ;
                     [ PROMPT <cPrompt> ] ;
                     [ SIZE <nWidth>, <nHeight> ] ;
                     [ WHEN <uWhen> ] ;
                     [ VALID <uValid> ] ;
                     [ UPDATE ]

@ <nRow>, <nCol> COMBOBOX [ <oCbx> VAR ] <cVar> ;
                     [ OF | WINDOW | DIALOG <oWnd> ] ;
                     [ PROMPTS | ITEMS <aItems> ] ;
                     [ SIZE <nWidth>, <nHeight> ] ;
                     [ WHEN <uWhen> ] ;
                     [ VALID <uValid> ] ;
                     [ UPDATE ]

@ <nRow>, <nCol> FOLDER [<oFolder>] ;
                     [ OF | WINDOW | DIALOG <oWnd> ] ;
                     [ PROMPT | PROMPTS | ITEMS <cPrompt,...> ] ;
                     [ SIZE <nWidth>, <nHeight> ] ;
                     [ UPDATE ]

@ <nRow>, <nCol> GET [ <oGet> VAR ] <uVar> ;
                     [ OF | WINDOW | DIALOG <oWnd> ] ;
                     [ SIZE <nWidth>, <nHeight> ] ;
                     [ WHEN <uWhen> ] ;
                     [ VALID <uValid> ] ;
                     [ MULTILINE | MEMO | TEXT ] ;
                     [ UPDATE ]

@ <nRow>, <nCol> GET [ <oGet> VAR ] <uVar> ;
```

```
                    [ PICTURE <cPicture> ] ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ SIZE <nWidth>, <nHeight> ] ;
                    [ WHEN <uWhen> ] ;
                    [ VALID <uValid> ] ;
                    [ UPDATE ]

@ <nRow>, <nCol> GROUP [ <oGroup> ] ;
                    [ LABEL | PROMPT <cText> ] ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ SIZE <nWidth>, <nHeight> ]

@ <nRow>, <nCol> IMAGE [ <oImg> ] ;
                    [ FILENAME | FILE | DISK <cFileName> ] ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ SIZE <nWidth>, <nHeight> ]
                    [ UPDATE ]

@ <nRow>, <nCol> LISTBOX [ <oLbx> VAR ] <cnVar> ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ PROMPT | PROMPTS | ITEMS <aItems> ] ;
                    [ SIZE <nWidth>, <nHeight> ] ;
                    [ WHEN <uWhen> ] ;
                    [ VALID <uValid> ] ;
                    [ UPDATE ]

@ <nRow>, <nCol> METER | PROGRESS [ <oMeter> VAR ] <nVar> ;
                    [ TOTAL <nTotal> ] ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ SIZE <nWidth>, <nHeight> ] ;
                    [ UPDATE ]

@ <nRow>, <nCol> RADIO [ <oRadMenu> VAR ] <nVar> ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ PROMPT | PROMPTS | ITEMS <acItems> ] ;
                    [ SIZE <nWidth>, <nHeight> ] ;
                    [ WHEN <uWhen> ] ;
                    [ VALID <uValid> ] ;
                    [ UPDATE ]

@ <nRow>, <nCol> SAY [ <oSay> PROMPT  | VAR ] <cText> ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ SIZE <nWidth>, <nHeight> ]
                    [ UPDATE ]

@ <nRow>, <nCol> SCROLLBAR [ <oSbr> ] ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ SIZE <nWidth>, <nHeight> ] ;
                    [ VERTICAL ] ;
                    [ HORIZONTAL ] ;
                    [ PIXEL ] ;
                    [ DOWN | ON DOWN <uDownAction> ] ;
                    [ UP | ON UP <uUpAction> ]

MENU [ <oObjMenu> ] [ POPUP> ]

     MENUITEM [ <oMenuItem> PROMPT ] [<cPrompt>] ;
                    [ ACTION <uAction> ] ;
                    [ RESOURCE | NAME | RESNAME <cResName> ]
     SEPARATOR

ENDMENU

ACTIVATE < MENU | POPUP> <oMenu> ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ AT <nRow>, <nCol> ]

SET < MESSAGE | MESSAGE BAR | MSGBAR> ;
                    [ OF | WINDOW | DIALOG <oWnd> ] ;
                    [ TO <cMsg> ] ;
                    [ UPDATE ]

PRINTER <oPrn> [ FROM USER ]
```

```
        PAGE

        ENDPAGE

    ENDPRINTER
```

Common Commands Clauses

@ <nRow>, <nCol>

Specifies the coordinates where the control is to be placed. If the PIXEL clause is used, then they are considered in pixels, if not, they simulate a console screen row and column dimensions. You may also use decimal numbers for those values.

OF | WINDOW | DIALOG <oWnd>

Specifies the container object of the control

SIZE <nWidth>, <nHeight>

Specifies the dimensions of the GUI object

UPDATE

Specifies whether the control should be updated when its container is updated (<oWnd>:Update()).

VALID <uAction>

Specifies an expression to evaluate when a control is about to lose focus. If false, then the focus remains on the control. For windows and dialog boxes, it controls the ability to close them.

WHEN <uAction>

Specifies an expression to evaluate to decide if the control is active or not. This clause is evaluated every time a control loses focus, and before all controls are displayed.

Classes:

FiveLinux Class Hierarchy:

Class TWindow
    + Class TControl
        + All GUI controls Classes

Class TBar

Method New()	Constructor

Class TButton

DATA bAction	The action to perform when the button is clicked
METHOD New()	Constructor
METHOD NewBar()	Constructor for ButtonBars
METHOD Click()	Called from HandleEvent()
METHOD HandleEvent()	Process GUI events

METHOD LostFocus()	Process the lost focus event
METHOD SetText()	Set the text of the button
METHOD GetText()	Retrieve the button text

Class TCheckBox

METHOD New()	Constructor
METHOD Click()	Called from HandleEvent()
METHOD HandleEvent()	Process GUI events
METHOD SetCheck()	Turn on/off the checkbox checked status
METHOD GetCheck()	Retrieve the checkbox checked status
METHOD SetText()	Set the text of the checkbox
METHOD GetText()	Retrieve the text of the checkbox

Class TClipboard

DATA hClipboard	An internal handle of the clipboard
METHOD New()	Constructor
METHOD Clear()	Clear the clipboard contents
METHOD SetText()	Set the clipboard text contents
METHOD GetText()	Retrieve the text from the clipboard

Class TComboBox

DATA aItems	The items displayed in the dropdown list
METHOD New()	Constructor
METHOD Change()	Called from HandleEvent()
METHOD HandleEvent()	Process GUI events
METHOD LostFocus()	Process the lost focus event
METHOD SetItems()	Set the combobox dropdown list items
METHOD SetText()	Set the combobox text
METHOD GetText()	Retrieve the combobox text

Class TControl

DATA bSetGet	A bSetGet codeblock to manage a related variable value
DATA bWhen	A codeblock to determine whether to activate or deactivate the control
DATA lUpdate	Update the control contents if its container is updated
METHOD LostFocus()	Process the lost focus event
METHOD HandleEvent()	virtual method

Class TDialog

DATA lModal	The execution waits until the dialog is closed
METHOD New()	Constructor
METHOD Activate()	Show and let the user interact with the dialog box
METHOD HandleEvent()	Process GUI events

Class TFolder

DATA aPrompts	The labels to display on each folder page
DATA aDialogs	An array of dialog boxes, one for each folder page
METHOD New()	Constructor
METHOD SetPrompts()	Set the pages' labels

Class TGet

DATA oGet	A Harbour GET object
METHOD New()	Constructor
METHOD HandleEvent()	Process GUI events

METHOD KeyPress()	Process key press events
METHOD LostFocus()	Process lost focus event
METHOD SetPos()	Set the cursor position
METHOD SetText()	Set the text of the GET
METHOD GetText()	Retrieve the GET text

Class TGroup

METHOD New()	Constructor
METHOD SetText()	Set the group label
METHOD GetText()	Retrieve the group label

Class TImage

DATA cFileName	The name of the file that holds the image
METHOD New()	Constructor
METHOD LoadImage()	Load an image from a file

Class TListBox

DATA aItems	The array of items displayed in the listbox
DATA nAt	The index of the selected item
METHOD New()	Constructor
METHOD HandleEvent()	Process GUI events
METHOD Change()	Called when an item is selected
METHOD SetItems()	Set the items of the listbox

Class TMenu

DATA hMenu	An internal handle of the menu
DATA aItems	An array of the menuitems
METHOD New()	Constructor
METHOD Activate()	Show a popup menu
METHOD Add()	Add a menuitem

METHOD Command()	Process a selected menuitem action

Class TMenuItem

DATA hMenuItem	An internal handle of the menuitem
DATA cPrompt	The menuitem text
DATA oPopup	A contained popup menu
DATA bAction	The action to execute when the menuitem is selected
DATA cResName	The GTK resource name
METHOD New()	Constructor
METHOD Add()	Add a popup object to the menuitem

Class TMultiGet

METHOD New()	Constructor for a multiline “edit” text box
METHOD HandleEvent()	Process GUI events
METHOD LostFocus()	Process the lost focus event
METHOD SetText()	Set the text of the multiline GET
METHOD GetText()	Retrieve the text of the multiline GET

Class TMsgBar

DATA cMsg	The msgbar shown text
METHOD New()	Constructor
METHOD SetText()	Set the msgbar text

Class TPrinter

DATA hJob	An internal handle of the printer job
DATA hGpc	An internal handle of the gnome print context
DATA nPage	The current built page
METHOD New()	Constructor

METHOD Choose()	Shows a dialog box to select the printer
METHOD End()	Start the printing
METHOD nWidth()	Retrieve the printer width dimension
METHOD nHeight()	Retrieve the printer height dimension
METHOD StartPage()	Start a new page
METHOD EndPage()	End the current page
METHOD SetPos()	Set the origin for the next action
METHOD Say()	Write a text at a certain location
METHOD Line()	Draw a line

Class TProgress

DATA nTotal	The total amount represented by the progress bar
METHOD New()	Constructor
METHOD Set()	Set the actual shown progress bar value
METHOD SetText()	Show a text on the progress bar
METHOD SetTotal()	Set the total amount represented by the progress bar

Class TRadio

DATA oRadMenu	The TRadMenu container object
METHOD New()	Constructor
METHOD Click()	Called when the radio is clicked
METHOD HandleEvent()	Process GUI events
METHOD lChecked()	Check if the radio is selected
METHOD SetCheck()	Check/Uncheck the radio
METHOD SetText()	Set the radio text
METHOD GetText()	Retrieve the radio text

Class TRadMenu

DATA aItems	An array with all the managed TRadio

	control objects
DATA bSetGet	A bSetGet codeblock to manage a related variable value
DATA bChange	A codeblock to evaluate when a radio item is selected
DATA hGroup	An internal handle of the radio group
DATA lUpdate	Update the control contents if its container is updated
METHOD New()	Constructor

Class TSay

METHOD New()	Constructor
METHOD SetText()	Set the text of the SAY
METHOD GetText()	Retrieve the text of the SAY

Class TScrollBar

DATA nValue	The actual value represented in the scrollbar
DATA bGoDown	A codeblock to evaluate when the scrollbar goes down
DATA bGoUp	A codeblock to evaluate when the scrollbar goes up
METHOD New()	Constructor
METHOD GoDown()	Called when the scrollbar goes down
METHOD GoUp()	Called when the scrollbar goes up
METHOD HandleEvent()	Process GUI events
METHOD SetRange()	Set the scrollbar represented range values
METHOD SetValue()	Set the currently represented scrollbar value
METHOD GetValue()	Retrieve the currently represented scrollbar value

Class TWBColumn

DATA cHeading	The heading text of the column
DATA bBlock	A codeblock to retrieve the column data to display
DATA nWidth	The width of the column
METHOD New()	Constructor

Class TWBrowse

WARNING: FiveLinux Class TWBrowse uses TWBColumn objects. Though its xBase syntax remains the same as FiveWin, this capability enhances the browse power. Class TWBrowse automatically creates such columns.

DATA aColumns	An array of column objects
DATA cAlias	The alias of a used workarea
DATA bSkip	A codeblock performed to skip n rows
DATA bChange	A codeblock to evaluate when a row is selected
DATA bGoTop	A codeblock to evaluate to go to the top
DATA bGoBottom	A codeblock to evaluate to go to the bottom
DATA bLogicLen	A codeblock to get the virtual row count of the entire current data set
DATA lHitTop	If the top most row has been reached
DATA lHitBottom	If the bottom most row has been reached
DATA nRowPos	The selected row from the visible ones
DATA nColPos	The left most visible column
DATA nLen	The value returned by bLogicLen
DATA oVScroll	The related vertical scrollbar object
DATA oHScroll	The related horizontal scrollbar object
METHOD New()	Constructor
METHOD AddColumn()	Add a column object
METHOD DrawHeaders()	Draw the browse headers

METHOD DrawLine()	Draw a browse row
METHOD DrawLines()	Draw all the visible browse rows
METHOD DrawRows()	Organize the DrawLines()
METHOD DrawSelect()	Draw the currently selected row
METHOD GoBottom()	Go to the bottom most record
METHOD GoDown()	Go down one row
METHOD GoLeft()	Scroll to the left
METHOD GoRight()	Scroll to the right
METHOD GoTop()	Go to the top most record
METHOD GoUp()	Go up one row
METHOD HandleEvent()	Process GUI events
METHOD KeyDown()	Called when a key is pressed
METHOD LButtonDown()	Called when the mouse is left clicked
METHOD nRowCount()	Return the number of visible rows
METHOD PageDown()	Go down one full page
METHOD PageUp()	Go up one full page
METHOD Paint()	Organize the painting of the browse
METHOD RButtonDown()	Called when the right mouse button is clicked
METHOD Skip()	Skip a certain amount of records

Class TWindow

DATA	hWnd	An internal handle of the window, dialog or control
DATA	oWnd	The container parent window or dialog
DATA	oMenu	The pulldown menu object if defined
DATA	oPopup	The popup menu object if defined
DATA	oMsgBar	The message bar object if defined
DATA	aControls	An array with all the child controls objects
DATA	bValid	A block to allow the close or loss of focus on controls
DATA	bLClicked	A codeblock to evaluate when the mouse is L clicked
DATA	bRClicked	A codeblock to evaluate when the mouse is

		R clicked
DATA	bReSized	A codeblock to evaluate when the window is resized
DATA	cargo	user defined cargo value
METHOD	New()	Constructor
METHOD	Activate()	Show and let the user interact with the window
METHOD	AddControl()	Automatically add a control to DATA aControls when a control is created as its child.
METHOD	AEvalWhen()	Evaluate the WHEN of each child control
METHOD	Center()	Center the window on the screen
METHOD	_cToolTip()	Set the tooltip of the window, dialog, control
METHOD	Disable()	Disable a window, dialog, control
METHOD	Enable()	Enable a window, dialog, control
METHOD	End()	Try to close a window or dialog
METHOD	HandleEvent()	Process GUI events
METHOD	Hide()	Hide a window, dialog or control
METHOD	LButtonDown()	Called when the mouse is left clicked
METHOD	lFocused()	Return .t. if the window, dialog or control is focused
METHOD	Link()	Internally used
METHOD	Maximize()	Maximize a window
METHOD	nLeft()	Retrieve the left position of the window, dialog or control
METHOD	nHeight()	Retrieve the height of the window, dialog or control
METHOD	nTop()	Retrieve the top position of the window, dialog or control
METHOD	nWidth()	Retrieve the width of the window, dialog or control
METHOD	RButtonDown()	Called when the mouse is right clicked

METHOD Refresh()	Force the repaint
METHOD ReSize()	Called when a window is resized
METHOD SetFocus()	Set the focus to this object
METHOD SetMenu()	Set a pulldown menu object
METHOD SetText()	Change the caption of the window or dialog
METHOD Show()	Make the window, dialog or control visible
METHOD Update()	Force the repaint of all child controls that have lUpdate set to true.

Implementing new GUI control Classes:

You may review FiveLinux classes/say.prg and winapi/says.c modules, as they are the simplest samples to be used to develop new classes that implement GTK+ controls not yet available in FiveLinux.

Glade provides a great help to create new control implementation, as it generates C source code where you can see the GTK+ functions used to create the controls.

To compile them use the following command lines:

```
harbour your.prg /n

gcc -c -o your.o -I./../harbour/include your.c

gcc -c -o yours.o -I./../harbour/include yours.c
```

Please notice that the OBJ files have the .o extension. To create your own libraries you may use the following command line:

```
ar rc ./libyours.a ./your.o
```

```
   ar rc ./libyours.a ./yours.o
```

Please notice that the libraries have the .a extension, and also that they use a lib prefix though such prefix is not used when linking your application. So from a libyours.a file, you will specify –lyours when linking.

To create new GTK+ controls from scratch, you may review classes/wbrowse.prg and classes/wbrowses.c.

Functions:

ASend()

Syntax: ASend(<aArray> ,<cMessage> [,<param>, ...])

Description: Sends an object oriented message to an array of objects.

ChooseColor()

Syntax: ChooseColor(<cTitle> [,<nColor>]) → nNewColor

Description: Displays a built-in color selection dialog box.

ChooseFont()

Syntax: ChooseFont(<cTitle> [,<cFont>]) → cNewFont

Description: Displays a built-in font selection dialog box.

cGetFile()

Description: Displays a built-in file selection dialog box.

cValToChar()

Syntax: cValToChar(<uValue>) → <cStrValue>

Description: Turns any value into its correspondent string equivalent.

GetFocus()

Syntax: GetFocus()→ <hWndFocused>

Description: Returns the hWnd handle of the currently focused window, dialog or control.

LogFile()

Syntax: LogFile(<cFileName>, <aValues>)

Description: Creates an ascii log file with the contents on an array of values.

MsgAlert()

Description: Displays an Alert GUI message on the screen.

MsgInfo()

Description: Displays an Information GUI message on the screen.

MsgYesNo()

Syntax: MsgYesNo(<cQuestion>) → lYesNo

Description: Displays a question GUI message on the screen, with two choices: yes and no.

OSend()

Syntax: OSend(<oObject> ,<cMessage> [,<param>, ...])

Description: Sends an object oriented message to an object.

SetExecutable()

Syntax: SetExecutable(<cFileName>) → nil

Description: Changes the file permissions to allow it to be executed as an application.

SysRefresh()

Description: Allows Linux and GTK+ to process pending system messages.

uValBlank()

Syntax: uValBlank(<uValue>) → <uEmptyValue>

Description: Turns any value into an empty one of the same type.

WinExec()

Syntax: WinExec(<cAppName> [,<cParam>]) → nil

Description: Executes an external application.

Implementing your own C source code:

You may review FiveLinux provided C source code to easily understand how to develop your own C routines.

To compile them use the following command line:

```
gcc -c -o file.o -I./../harbour/include file.c
```

Please notice that the OBJ files have the .o extension. To create your own libraries you may use the following command line:

```
ar rc ./libyours.a ./file.o
```

Please notice that the libraries have the .a extension, and also that they use a lib prefix though such prefix is not used when linking your application. So from a libyours.a file, you will specify –lyours when linking.

FiveLinux samples

tutor01.prg	Typical Hello world message
tutor02.prg	A simple window with a VALID clause
tutor03.prg	A simple window with a button and a VALID clause
tutor04.prg	window, menu, buttonbar and control sample
testbrow.prg	Testing FiveLinux browses
testdlg.prg	Testing dialog boxes
testdlgs.prg	Testing built-in dialog boxes
testclip.prg	Testing the clipboard
testerr.prg	Testing FiveLinux error system
testfold.prg	Testing the folders
testpop.prg	Testing popup menus
testprn.prg	Testing the printer object
testrad.prg	Testing radios menus and multiline GETs
yelp.prg	Testing gnome help system

FiveWin components not available on FiveLinux yet

The following FiveWin components are not available on FiveLinux yet. We do expect to have them available for future releases:

- INI file management.
- MDI Windows: GTK+ does not provide it as a standard, we may simulate them in next releases.
- The Printing Preview: Metafiles are not yet available. We are looking for a way to implement them.
- Resources as RC files: We are working to implement a way to use them as in Windows. We do expect to have them ready for next FiveLinux release. Actually you have to develop your code using @ ..., ... commands, though we are going to provide an utility to read glade files and turn them into @ ..., ... commands.

Technical support and upgrades

Please follow these steps when you do need technical support:

First post a message on our news server. Keep in mind that some other users may have experienced the same problem and they may easily help you. Our actual news server is located at news://news.ozs.com, but we may change it soon into news://news.fivetechsoft.com.

In case you don't get an answer on the news server, please send an email to alinares@fivetechsoft.com. Please don't send us any emails if you have not posted your question on our news server previously. Thanks!

Important: Please try to provide a small and easy to build sample that reproduces the error. Our technical support department may quickly answer you if you cooperate in this way.

We appreciate if you want to send us any fix or enhancement. We will review it, we will comment it to you and we will include it in the next build of our products. Thanks!

You may visit www.fivetechsoft.com news section to obtain FiveLinux upgrade info.