

## *HwGUI – Manual de Referência para Versão 2.07*

Alexander S.Kresin

Maio 2003

Tradução para o Português do Brasil

Paulo Sóstenes

psostenes@yahoo.com.br

- 1 Introdução
  - ◆ 1.1 O que é a HwGUI
  - ◆ 2 A História da HwGUI
- 2 Instalação da HwGUI
  - ◆ 2.1 O Pacote HwGUI
  - ◆ 2.2 Como Montar a Biblioteca HwGUI
  - ◆ 2.3 Como Montar os Exemplos HwGUI
- 3 Como Utilizar a HwGUI
  - ◆ 3.1 O Primeiro Aplicativo HwGUI
- 4 Comandos
  - ◆ 4.1 Manuseando Janelas e Caixas de Diálogos
    - ◇ 4.1.1 INIT WINDOW
    - ◇ 4.1.2 INIT DIALOG
    - ◇ 4.1.3 ACTIVATE WINDOW
    - ◇ 4.1.4 ACTIVATE DIALOG
    - ◇ 4.1.5 DIALOG ACTIONS OF
  - ◆ 4.2 Manipulando Menus
    - ◇ 4.2.1 MENU
    - ◇ 4.2.2 ENDMENU
    - ◇ 4.2.3 MENUITEM
    - ◇ 4.2.4 SEPARATOR
    - ◇ 4.2.5 MENU FROM RESOURCE OF
    - ◇ 4.2.6 CONTEXT MENU
  - ◆ 4.3 Manipulando controles
    - ◇ 4.3.1 ADD STATUS TO
    - ◇ 4.3.2 @ <x>.<y> SAY
    - ◇ 4.3.3 @ <x>.<y> EDITBOX
    - ◇ 4.3.4 REDEFINE EDITBOX
    - ◇ 4.3.5 @ <x>.<y> BUTTON
    - ◇ 4.3.6 REDEFINE BUTTON
    - ◇ 4.3.7 @ <x>.<y> CHECKBOX
    - ◇ 4.3.8 REDEFINE CHECKBOX
    - ◇ 4.3.9 @ <x>.<y> COMBOBOX
    - ◇ 4.3.10 REDEFINE COMBOBOX
    - ◇ 4.3.11 @ <x>.<y> RADIOBUTTON
    - ◇ 4.3.12 REDEFINE RADIOBUTTON
    - ◇ 4.3.13 RADIOGROUP
    - ◇ 4.3.14 END RADIOGROUP
    - ◇ 4.3.15 @ <x>.<y> PANEL
    - ◇ 4.3.16 REDEFINE PANEL
    - ◇ 4.3.17 @ <x>.<y> BROWSE
    - ◇ 4.3.18 REDEFINE BROWSE

- ◇ 4.3.19 ADD COLUMN
  - ◇ 4.3.20 @ <x>.<y> OWNERBUTTON
  - ◇ 4.3.21 REDEFINE OWNERBUTTON
  - ◇ 4.3.22 @ <x>.<y> GROUPBOX
  - ◇ 4.3.23 @ <x>.<y> DATEPICKER
  - ◇ 4.3.24 @ <x>.<y> UPDOWN
  - ◇ 4.3.25 @ <x>.<y> TAB
  - ◇ 4.3.26 BEGIN PAGE
  - ◇ 4.3.27 END PAGE
  - ◇ 4.3.28 @ <x>.<y> TREE
  - ◇ 4.3.29 INSERT NODE
  - ◇ 4.3.30 SET TIMER
- ◆ 4.4 Sistema Get
  - ◇ 4.4.1 @ <x>.<y> GET
  - ◇ 4.4.2 REDEFINE GET
  - ◇ 4.4.3 @ <x>.<y> GET CHECKBOX
  - ◇ 4.4.4 REDEFINE GET CHECKBOX
  - ◇ 4.4.5 @ <x>.<y> GET COMBOBOX
  - ◇ 4.4.6 REDEFINE GET COMBOBOX
  - ◇ 4.4.7 GET RADIOGROUP
  - ◇ 4.4.8 @ <x>.<y> GET DATEPICKER
  - ◇ 4.4.9 @ <x>.<y> GET UPDOWN
- 5 Funções
- 6 Classes
  - ◆ 6.1 Hierarquia de Classe
  - ◆ 6.2 HObject
  - ◆ 6.3 HCustomWindow
  - ◆ 6.4 HWindow
  - ◆ 6.5 HDialog
  - ◆ 6.6 HControl
  - ◆ 6.8 HBrowse
  - ◆ 6.9 HColumn
  - ◆ 6.10 HPanel
  - ◆ 6.11 HOwnerButton
  - ◆ 6.12 HStatus
  - ◆ 6.13 HStatic
  - ◆ 6.14 HButton
  - ◆ 6.15 HEdit
  - ◆ 6.16 HChexBox
  - ◆ 6.17 HRadioButton
  - ◆ 6.18 HCombobox
  - ◆ 6.19 HGroup
  - ◆ 6.20 HFont
  - ◆ 6.21 HPen
  - ◆ 6.22 HBrush
  - ◆ 6.23 HBitmap
  - ◆ 6.24 HPrinter
  - ◆ 6.25 HDatePicker
  - ◆ 6.26 HUpDown
  - ◆ 6.27 HTab
  - ◆ 6.28 HTree

- ♦ 6.29 HTreeNode
- ♦ 6.30 HTimer
- 7 Integração com o QHTM
  - ♦ 7.1 Visão Geral
  - ♦ 7.2 Comandos
  - ♦ 7.3 Funções
- 8 Licença
- 9 Sobre o Autor
- 10 Sobre a Tradução

## **1 Introdução**

### **1.1 O Que é HwGUI**

A HwGUI é uma biblioteca para o Harbour/xHarbour, com o fim de criar aplicativos GUI sobre a plataforma Win32. É baseada em chamadas diretas à API Win32, por isso **não pode ser usada em outros sistemas operacionais**. Em tempo de desenvolvimento a HwGUI procura ocultar do desenvolvedor – o programador Harbour, os detalhes técnicos de chamadas à API e monta um conjunto de comandos e funções, o que permitiu, assim, criar e gerenciar objetos GUI facilmente.

### **1.2 A História da HwGUI**

Iniciei o trabalho sobre a HwGUI em Agosto 2001 e a primeira versão foi liberada aos dias 21 de Agosto. Minha intenção inicial foi criar uma pequena e rápida LIB GUI principalmente para atender minhas necessidades. E já em Outubro escrevi minha primeira aplicação pequena e estável com a HwGUI, ela leu as bases de dados, criou e gerenciou registros, gerou alguns documentos e os enviou por fax., em um sistema de demonstração.

Iniciamente, até a liberação da HwGUI 1.3 não utilizava o paradigma OOP – e a declarei como uma das características da HwGUI. Minha motivação principal era rapidez e estabilidade. A implementação de classes do Harbour no momento tinha alguns bugs e não desejava aumentar meus problemas. E, nesse sentido, acessar variáveis de objeto é mais lento do que acessar elementos de array. A OOP é um nível adicional e usá-la reduz a performance dos aplicativos.

Mas depois tive de chegar a decisão de fazer a HwGUI baseada em OOP – para simplificar a interface para o usuário e fazê-la melhor estruturada e mais cômoda. Então uso de classes foi a partir da liberação da HwGUI 2.0.

## **2 Instalação da HwGUI**

### **2.1 O Pacote HwGUI**

A HwGUI é distribuída em um pacote zip. Atualmente não se utiliza qualquer utilitário de instalação, você precisa apenas descompactá-la no lugar que desejar. O pacote zip inclui os seguintes arquivos e pastas:

```
make_b32.bat
makefile.bc
make_vc.bat
makefile.vc
license.txt
```

install.txt	
whatsnew.txt	
DOC	– Pasta com a Documentação
INCLUDE	– Pasta com os Arquivos de Cabeçalhos da HwGUI
LIB	– Pasta para a Lib HwGUI
OBJ	
SAMPLES	– Pasta com os Exemplos HwGUI
SOURCE	– Pasta com os Arquivos Fonte da Hwgui

## **2.2 Como Montar a Biblioteca HwGUI**

Antes de gerar a biblioteca você deve checar os valores da variável de ambiente HRB\_DIR no arquivo HWGUI/makefile.bc. Ela deve apontar para a pasta onde sua cópia do Harbour está instalada. Então, execute o make\_b32.bat – e este montará duas bibliotecas – **hwgui.lib** e **procmisc.lib**. Isso é tudo!

## **2.3 Como Montar os Exemplos da HwGUI**

### **3 Como usar a HwGUI**

#### **3.1 O Primeiro Aplicativo HwGUI**

```

Function Main
Local oMainWnd, oFont
Local aCombo := {"Primeiro","Segundo" }

    PREPARE FONT oFont NAME "MS Sans Serif" WIDTH 0 HEIGHT -13
    INIT WINDOW oMainWnd TITLE "Exemplo" ;
        FONT oFont ;
        ON EXIT {||MsgYesNo("Deseja Realmente Finalizar ?") }

    @ 20,10 EDITBOX "Olá, Mundo!" ;
        SIZE 200,30 ;
        STYLE WS_DLGFRAME+ES_LEFT

    @ 270,10 COMBOBOX aCombo ;
        STYLE WS_TABSTOP ;
        SIZE 100, 150 TOOLTIP "Combobox"

    @ 120,60 BUTTON "Fecha" ;
        SIZE 150,30 ;
        ON CLICK {||EndWindow()}

    MENU OF oMainWnd
        MENUITEM "Sobre" ACTION MsgInfo("Primeiro Aplicativo
HwGUI ")
    ENDMENU

    ACTIVATE WINDOW oMainWnd

```

Return

A primeira coisa que desejará fazer, acredito eu, será criar a janela principal. O melhor modo de fazer isto é com o comando INIT WINDOW. Neste comando você pode definir a posição inicial eo tamanho da janela, seu estilo, ícone e cor de fundo. Você pode configurar também os eventos manipuladores – codeblocks, que são avaliados por outros eventos ( INIT, EXIT, PAINT, alterando seu SIZE, GETFOCUS, LOSTFOCUS, etc ).

Então você precisa definir controles para a janela e o menu principal (comandos MENU ... ENDMENU ), e, por último, ativar a janela, ( ACTIVATE WINDOW ) e exibí-la na tela. Analize o exemplo a seguir:

```
PREPARE FONT oFont NAME "MS Sans Serif" WIDTH 0 HEIGHT -13
```

No início, criamos o objeto font para a janela principal. A HwGUI trabalha deste modo, se a fonte não for definida para um controle, este controle usa a fonte definida pela janela "Pai", ou a janela sobre a qual está posicionado.

```
INIT WINDOW oMainWnd MAIN TITLE "Exemplo" ;  
    FONT oFont ;  
    ON EXIT {||MsgYesNo("Deseja Realmente Finalizar ?")}
```

Este comando cria uma janela principal com o título "Exemplo" e com uma fonte previamente definida. A cláusula ON EXIT causará a exibição da caixa de mensagem, o usuário deverá escolher "Sim" para terminar a execução do aplicativo.

```
@ 20,10 EDITBOX "Olá, Mundo!" ;  
    SIZE 200,30 ;  
    STYLE WS_DLGFRAME+ES_LEFT  
@ 270,10 COMBOBOX aCombo ;  
    STYLE WS_TABSTOP ;  
    SIZE 100, 150 TOOLTIP "Combobox"  
@ 120,60 BUTTON "Fecha" ;  
    SIZE 150,30 ;  
    ON CLICK {||EndWindow()}
```

Os comandos acima criam os respectivos controles – EDIT, COMBOBOX e um Push BUTTON (ou botão de Pressão). O ComboBox é inicializado com uma array aCombo, que foi declarada previamente. O tem um manipulador de evento definido – EndWindow.

```
MENU OF oMainWnd  
    MENUITEM "Sobre" ACTION MsgInfo("Primeiro Aplicativo HwGUI")  
ENDMENU
```

Estes comandos criam o menu principal, o qual inclui apenas o item "Sobre".

```
ACTIVATE WINDOW oMainWnd
```

E, por fim, este último comando ativa a janela principal. Ela aparece na tela com menu e todos os controles definidos.

## 4 Comandos

### 4.1 Manuseando Janelas e Caixas de Diálogos

#### 4.1.1 INIT WINDOW

Este comando cria uma janela – MAIN, MDI ou MDI CHILD. Para forçar sua exibição sobre a tela você precisará ativar esta opção posteriormente. Isto é possível com o comando ACTIVATE WINDOW ou com o método Activate() do objeto WINDOW.

INIT WINDOW <oWnd>

```
[ MAIN ]
[ MDI ]
[ MDICHILD ]
[ TITLE <cTitle> ]
[ AT <x>, <y> ]
[ SIZE <width>, <height> ]
[ ICON <ico> ]
[ COLOR <clr> ]
[ STYLE <nStyle> ]
[ FONT <oFont> ]
[ MENU <cMenu> ]
[ MENUPOS <nPos> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bPaint> ]
[ ON GETFOCUS <bGfocus> ]
[ ON LOSTFOCUS <bLfocus> ]
[ ON OTHER MESSAGES <bOther> ]
[ ON EXIT <bExit> ]
```

As cláusulas MAIN, MDI, MDICHILD especificam o tipo de janela, e apenas uma destas cláusulas pode ser usada neste comando.

**cTitle** – O título da janela;

**x,y,width,height** – As coordenadas **x,y** são respectivamente a **Linha** e a **Coluna** em pixels, que definem a margem esquerda e o topo da janela, e em seguida sua **largura** e **altura**;

**ico** – O nome do Ícone no arquivo de recursos (resource);

**clr** – A cor de fundo da janela;

**nStyle** – O estilo da janela;

**oFont** – O objeto HFont previamente definido;

**cMenu** – O nome da declaração de menu arquivo de recursos (se você definir um menu nos recursos);

**nPos** – Identifica a janela com menu de opções usado para controlar janelas MDI "filhas". Quando janelas filhas são criadas, o aplicativo adiciona os títulos delas como opções para este menu. O usuário pode então ativar uma janela filha pela escolha do seu título no menu de opções.

**bSize** – Um codeblock, que será avaliado durante o redimensionamento da janela;

**bPaint** – Um codeblock, que será avaliado durante o desenho da janela;

**bGFocus** – Um codeblock, que será avaliado quando a janela estiver em foco;

**bLFocus** – Um codeblock, que será avaliado quando a janela perder o foco;

**bExit** – Um codeblock, que será avaliado quando a janela é fechada;

**bOther** – Um codeblock, que será avaliado por todas as outras mensagens – você pode escrever seu próprio

procedimento para manipulação de mensagens;

#### 4.1.2 INIT DIALOG

Este comando cria uma caixa de diálogo. Para forçar sua exibição sobre a tela você precisará ativar esta opção posteriormente. Isto é possível com o comando ACTIVATE DIALOG ou com o método Activate() do objeto janela.

INIT DIALOG <oWnd>

```
[ FROM RESOURCE ]  
[ TITLE <cTitle> ]  
[ AT <x>, <y> ]  
[ SIZE <width>, <height> ]  
[ STYLE <nStyle> ]  
[ FONT <oFont> ]  
[ CLIPPER ]  
[ ON INIT <bInit> ]  
[ ON SIZE <bSize> ]  
[ ON PAINT <bPaint> ]  
[ ON GETFOCUS <bGfocus> ]  
[ ON LOSTFOCUS <bLfocus> ]  
[ ON OTHER MESSAGES <bOther> ]  
[ ON EXIT <bExit> ]
```

Todas as cláusulas são idênticas as do comando INIT WINDOW, exceto numa cláusula específica – FROM RESOURCE – que será usado quando desejar **criar uma caixa de diálogo de um arquivo de recursos**.

#### 4.1.3 ACTIVATE WINDOW

ACTIVATE WINDOW <oWnd>

Este comando é igual à chamada 'oWnd:Activate()'. Exibe uma janela previamente definida e inicia o processo de mensagens para ela.

#### 4.1.4 ACTIVATE DIALOG

ACTIVATE DIALOG <oDlg> [ NOMODAL ]

Este comando é igual à chamada 'oDlg:Activate()'. Exibe uma caixa de diálogo previamente definida e inicia o processo de mensagens para ela. Por padrão, cria uma caixa de diálogo modal. É possível criar uma caixa de diálogo não modal, especificando a cláusula NOMODAL neste comando.

#### 4.1.5 DIALOG ACTIONS OF

DIALOG ACTIONS OF <oWnd>

```
ON <id1>,<id2> ACTION <b1>  
[ ON <idn1>,<idn2> ACTION <bn> ]
```

Este comando é usado por usuários, que estão familiarizados com a API Win32. A janela obtém a mensagem WM\_COMMAND quando o usuário escolhe um item no menu de opções, quando o controle envia uma mensagem de notificação para a janela pai, ou quando uma tecla de atalho é interpretada.

É possível especificar neste comando o evento, a origem do evento e a ação apropriada,

id1 ...idn1 – o código evento,

id1 ... idn2 – o identificador do controle,

b1 ... bn – o codeblock.

Dê uma olhada nos exemplos e veja como este comando pode ser usado.

## **4.2 Manipulando Menus**

### **4.2.1 MENU**

MENU [ OF <oWnd> ] [ ID <nId> ] [ TITLE <cTitle> ]

### **4.2.2 ENDMENU**

ENDMENU

### **4.2.3 MENUITEM**

MENUITEM <item> [ ID <nId> ] ACTION <act>

### **4.2.4 SEPARATOR**

SEPARATOR

### **4.2.5 MENU FROM RESOURCE OF**

MENU FROM RESOURCE OF <oWnd>

ON <id1> ACTION <b1>

[ ON <idn> ACTION <bn> ]

Este comando é usado se especificar um menu em um arquivo de recurso, definindo ações para cada item do menu.

### **4.2.6 CONTEXT MENU**

CONTEXT MENU <oMenu>

## **4.3 Manipulando Controles**

Todos os comandos a seguir criam as instâncias de classes, elas são pré-processadas no método New() ou Redefine() da respectiva classe.

### **4.3.1 ADD STATUS TO**

ADD STATUS [ TO <oWnd> ]



```

[ ID <nId> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ STYLE <nStyle> ]
[ FONT <oFont> ]
[ PARTS <aparts,...> ]

```

#### 4.3.2 @ <x>.<y> SAY

@ <x>.<y> SAY [ <oSay> CAPTION ] <caption>

```

[ OF <oWnd> ];
[ ID <nId> ]
[ SIZE <width>, <height> ]
[ COLOR <tcOLOR> ]
[ BACKCOLOR <bcolor> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ STYLE <nStyle> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

#### 4.3.3 @ <x>.<y> EDITBOX

@ <x>.<y> EDITBOX [ <oEdit> CAPTION ] <caption>

```

[ OF <oWnd> ];
[ ID <nId> ]
[ SIZE <width>, <height> ]
[ COLOR <tcOLOR> ]
[ BACKCOLOR <bcolor> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ ON GETFOCUS <bGfocus> ]
[ ON LOSTFOCUS <bLfocus> ]
[ STYLE <nStyle> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

#### 4.3.4 REDEFINE EDITBOX

REDEFINE EDITBOX <oEdit>

```

[ OF <oWnd> ]
ID <nId>
[ COLOR <tcOLOR> ]
[ BACKCOLOR <bcolor> ]
[ ON INIT <bInit> ]

```

```

[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ ON GETFOCUS <bGfocus> ]
[ ON LOSTFOCUS <bLfocus> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

#### 4.3.5 @ <x>,<y> BUTTON

@ <x>,<y> BUTTON [ <oButton> CAPTION ] <caption>

```

[ OF <oWnd> ];
[ ID <nId> ]
[ SIZE <width>, <height> ]
[ COLOR <tcOLOR> ]
[ BACKCOLOR <bcolor> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ ON CLICK <bClick> ]
[ STYLE <nStyle> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

#### 2 4.3.6 REDEFINE BUTTON

REDEFINE BUTTON <oButton>

```

[ OF <oWnd> ]
ID <nId>
[ COLOR <tcOLOR> ]
[ BACKCOLOR <bcolor> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ ON CLICK <bClick> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

#### 4.3.7 @ <x>,<y> CHECKBOX

@ <x>,<y> CHECKBOX [ <oCheck> CAPTION ] <caption>

```

[ OF <oWnd> ];
[ ID <nId> ]
[ SIZE <width>, <height> ]
[ COLOR <tcOLOR> ]
[ BACKCOLOR <bcolor> ]
[ INIT <lInit> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]

```

```

[ ON PAINT <bDraw> ]
[ STYLE <nStyle> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

#### 4.3.8 REDEFINE CHECKBOX

REDEFINE CHECKBOX <oCheck>

```

[ OF <oWnd> ]
ID <nId>
[ COLOR <tcOLOR> ]
[ BACKCOLOR <bcolor> ]
[ INIT <lInit> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

#### 4.3.9 @ <x>.<y> COMBOBOX

@ <x>.<y> COMBOBOX [ <oCombo> ITEMS ] <aItems>

```

[ OF <oWnd> ];
[ ID <nId> ]
[ SIZE <width>, <height> ]
[ INIT <nInit> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ ON CHANGE <bChange> ]
[ STYLE <nStyle> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

#### 4.3.10 REDEFINE COMBOBOX

REDEFINE COMBOBOX [ <oCombo> ITEMS ] <aItems>

```

[ OF <oWnd> ]
ID <nId>
[ INIT <nInit> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ ON CHANGE <bChange> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

#### 4.3.11 @ <x>.<y> RADIOBUTTON

@ <x>,<y> RADIOBUTTON [ <oRadio> CAPTION ] <caption>

[ OF <oWnd> ];  
[ ID <nId> ]  
[ SIZE <width>, <height> ]  
[ COLOR <tcolor> ]  
[ BACKCOLOR <bcolor> ]  
[ ON INIT <bInit> ]  
[ ON SIZE <bSize> ]  
[ ON PAINT <bDraw> ]  
[ STYLE <nStyle> ]  
[ FONT <oFont> ]  
[ TOOLTIP <ctoolt> ]

#### 4.3.12 REDEFINE RADIOBUTTON

REDEFINE RADIOBUTTON <oRadio>

[ OF <oWnd> ]  
ID <nId>  
[ COLOR <tcolor> ]  
[ BACKCOLOR <bcolor> ]  
[ ON INIT <bInit> ]  
[ ON SIZE <bSize> ]  
[ ON PAINT <bDraw> ]  
[ FONT <oFont> ]  
[ TOOLTIP <ctoolt> ]

#### 4.3.13 RADIOGROUP

RADIOGROUP

#### 4.3.14 END RADIOGROUP

END RADIOGROUP [ SELECTED <nSel> ]

#### 4.3.15 @ <x>,<y> PANEL

@ <x>,<y> PANEL <oPanel>

[ OF <oWnd> ];  
[ ID <nId> ]  
[ SIZE <width>, <height> ]  
[ ON INIT <bInit> ]  
[ ON SIZE <bSize> ]  
[ ON PAINT <bDraw> ]  
[ STYLE <nStyle> ]

#### 4.3.16 REDEFINE PANEL

REDEFINE PANEL <oCheck>

```

[ OF <oWnd> ]
ID <nId>
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]

```

#### 4.3.17 @ <x>.<y> BROWSE

@ <x>.<y> BROWSE <oBrowse>

```

[ ARRAY ]
[ DATABASE ]
[ OF <oWnd> ];
[ ID <nId> ]
[ SIZE <width>, <height> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ ON CLICK <bClick> ]
[ ON GETFOCUS <bGetFocus> ]
[ ON LOSTFOCUS <bLostFocus> ]
[ STYLE <nStyle> ]
[ FONT <oFont> ]

```

#### 4.3.18 REDEFINE BROWSE

REDEFINE BROWSE <oBrowse>

```

[ ARRAY ]
[ DATABASE ]
[ OF <oWnd> ]
ID <nId>
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON PAINT <bDraw> ]
[ ON CLICK <bClick> ]
[ ON GETFOCUS <bGetFocus> ]
[ ON LOSTFOCUS <bLostFocus> ]
[ FONT <oFont> ]

```

#### 4.3.19 ADD COLUMN

ADD COLUMN <block> <

```

TO <oBrowse>
[ HEADER <cTitle> ]
[ TYPE <type> ]
[ LEN <length> ]
[ DEC <dec> ]
[ <lEdit: EDITABLE> ]

```

#### 4.3.20 @ <x>.<y> OWNERBUTTON

```
@ <x>.<y> OWNERBUTTON <oOwnBtn>

    [ OF <oWnd> ];
    [ ID <nId> ]
    [ SIZE <width>, <height> ]
    [ ON INIT <bInit> ]
    [ ON SIZE <bSize> ]
    [ ON PAINT <bDraw> ]
    [ ON CLICK <bClick> ]
    [ STYLE <nStyle> ]
    [ FLAT ]
    [ TEXT <cText>
        [ COLOR <color> ] [ FONT <font> ]
        [ COORDINATES <xt>, <yt>, <widthtt>, <heightt> ]
    ]
    [ BITMAP <bmp> [ FROM RESOURCE ]
        [ COORDINATES <xb>, <yb>, <widthtb>, <heightb> ]
    ]
]
```

#### 4.3.21 REDEFINE OWNERBUTTON

```
REDEFINE OWNERBUTTON <oOwnBtn>

    [ OF <oWnd> ]
    ID <nId>
    [ ON INIT <bInit> ]
    [ ON SIZE <bSize> ]
    [ ON PAINT <bDraw> ]
    [ ON CLICK <bClick> ]
    [ STYLE <nStyle> ]
    [ FLAT ]
    [ TEXT <cText>
        [ COLOR <color> ] [ FONT <font> ]
        [ COORDINATES <xt>, <yt>, <widthtt>, <heightt> ]
    ]
    [ BITMAP <bmp> [ FROM RESOURCE ]
        [ COORDINATES <xb>, <yb>, <widthtb>, <heightb> ]
    ]
]
```

#### 4.3.22 @ <x>.<y> GROUPBOX

```
@ <x>.<y> GROUPBOX [ <oGroup> CAPTION ] <caption>

    [ OF <oWnd> ];
    [ ID <nId> ]
    [ SIZE <width>, <height> ]
    [ COLOR <tcolor> ]
    [ BACKCOLOR <bcolor> ]
    [ ON INIT <bInit> ]
```

[ ON SIZE <bSize> ]  
 [ ON PAINT <bDraw> ]  
 [ STYLE <nStyle> ]  
 [ FONT <oFont> ]

#### 4.3.23 @ <x>,<y> DATEPICKER

@ <x>,<y> DATEPICKER [ <oPicker> CAPTION ] <caption>

[ OF <oWnd> ]  
 ID <nId>  
 [ SIZE <width>, <height> ]  
 [ COLOR <tcolor> ]  
 [ BACKCOLOR <bcolor> ]  
 [ INIT <value> ]  
 [ ON INIT <bInit> ]  
 [ ON GETFOCUS <bGetFocus> ]  
 [ ON LOSTFOCUS <bLostFocus> ]  
 [ STYLE <nStyle> ]  
 [ FONT <oFont> ]  
 [ TOOLTIP <ctoolt> ]

#### 4.3.24 @ <x>,<y> UPDOWN

@ <x>,<y> UPDOWN [ <oUpDown> INIT ] <nInit>

RANGE <nLower>, <nUpper>  
 [ OF <oWnd> ]  
 [ ID <nId> ]  
 [ SIZE <width>, <height> ]  
 [ WIDTH <nUpdWidth> ]  
 [ COLOR <tcolor> ]  
 [ BACKCOLOR <bcolor> ]  
 [ ON INIT <bInit> ]  
 [ ON SIZE <bSize> ]  
 [ ON PAINT <bDraw> ]  
 [ ON GETFOCUS <bGfocus> ]  
 [ ON LOSTFOCUS <bLfocus> ]  
 [ STYLE <nStyle> ]  
 [ FONT <oFont> ]  
 [ TOOLTIP <ctoolt> ]

#### 4.3.25 @ <x>,<y> TAB

@ <x>,<y> TAB [ <oTab> ITEMS ] <aTabs>

[ OF <oWnd> ]  
 [ ID <nId> ]  
 [ SIZE <width>, <height> ]  
 [ ON INIT <bInit> ]  
 [ ON SIZE <bSize> ]

[ ON PAINT <bDraw> ]  
[ STYLE <nStyle> ]  
[ FONT <oFont> ]

#### 4.3.26 BEGIN PAGE

BEGIN PAGE <cname> OF <oTab>

#### 4.3.27 END PAGE

END PAGE OF <oTab>

#### 4.3.28 @ <x>,<y> TREE

@ <x>,<y> TREE [ <oTree> ]  
  
[ OF <oWnd> ]  
[ ID <nId> ]  
[ SIZE <width>, <height> ]  
[ FONT <oFont> ]  
[ ON INIT <bInit> ]  
[ ON SIZE <bSize> ]  
[ STYLE <nStyle> ]  
[ BITMAP <aBmp> [<res: FROM RESOURCE>] ]

#### 4.3.29 INSERT NODE

INSERT NODE [ <oNode> TITLE ] <cTitle>  
  
TO <oTree>  
[ AFTER <oPrev> ]  
[ BITMAP <aBmp> ]  
[ ON CLICK <bClick> ]

#### 4.3.30 SET TIMER

SET TIMER [ <oTimer> ]  
  
[ OF <oWnd> ]  
[ ID <id> ]  
VALUE <value>  
ACTION <bAction>

### **4.4 Sistema Get**

O sistema Get system não usa uma classe especial, algo parecido com o HGet. Ele usa a mesma classe ( HEdit, HCheckBox, etc. ), como comandos comuns para criar os controles, adicionando apenas alguns novos comportamentos.

#### 4.4.1 @ <x>,<y> GET



@ <x>,<y> GET [ <oEdit> VAR ] <vari>

[ OF <oWnd> ]  
[ ID <nId> ]  
[ SIZE <width>, <height> ]  
[ COLOR <tcolor> ]  
[ BACKCOLOR <bcolor> ]  
[ PICTURE <cPicture> ]  
[ WHEN <bWhen> ]  
[ VALID <bValid> ]  
[ STYLE <nStyle> ]  
[ FONT <oFont> ]  
[ TOOLTIP <ctoolt> ]

#### 4.4.2 REDEFINE GET

REDEFINE GET [ <oEdit> VAR ] <vari>

[ OF <oWnd> ]  
ID <nId>  
[ COLOR <tcolor> ]  
[ BACKCOLOR <bcolor> ]  
[ WHEN <bWhen> ]  
[ VALID <bValid> ]  
[ FONT <oFont> ]  
[ TOOLTIP <ctoolt> ]

#### 4.4.3 @ <x>,<y> GET CHECKBOX

@ <x>,<y> GET CHECKBOX [ <oEdit> VAR ] <vari>

[ OF <oWnd> ]  
[ ID <nId> ]  
[ SIZE <width>, <height> ]  
[ COLOR <tcolor> ]  
[ BACKCOLOR <bcolor> ]  
[ <valid: VALID,ON CLICK> <bValid> ]  
[ STYLE <nStyle> ]  
[ FONT <oFont> ]  
[ TOOLTIP <ctoolt> ]

#### 4.4.4 REDEFINE GET CHECKBOX

REDEFINE GET CHECKBOX [ <oEdit> VAR ] <vari>

[ OF <oWnd>]  
ID <nId>  
[ COLOR <tcolor> ]  
[ BACKCOLOR <bcolor> ]  
[ <valid: VALID,ON CLICK> <bValid> ]  
[ FONT <oFont> ]

[ TOOLTIP <ctoolt> ]

#### 4.4.5 @ <x>.<y> GET COMBOBOX

@ <x>.<y> GET COMBOBOX [ <oCombo> VAR ] <vari>

ITEMS ] <aItems>  
[ OF <oWnd> ]  
[ ID <nId> ]  
[ SIZE <width>, <height> ]  
[ ON CHANGE <bChange> ]  
[ STYLE <nStyle> ]  
[ FONT <oFont> ]  
[ TOOLTIP <ctoolt> ]

#### 4.4.6 REDEFINE GET COMBOBOX

REDEFINE GET COMBOBOX [ <oCombo> VAR ] <vari>

ITEMS ] <aItems>  
[ OF <oWnd> ]  
ID <nId>  
[ ON CHANGE <bChange> ]  
[ FONT <oFont> ]  
[ TOOLTIP <ctoolt> ]

#### 4.4.7 GET RADIOGROUP

GET RADIOGROUP <vari>

#### 4.4.8 @ <x>.<y> GET DATEPICKER

@ <x>.<y> GET DATEPICKER [ <oPick> VAR ] <vari>

[ OF <oWnd>]  
[ ID <nId> ]  
[ SIZE <width>, <height> ]  
[ COLOR <tcolor> ]  
[ BACKCOLOR <bcolor> ]  
[ WHEN <bWhen> ]  
[ VALID <bValid> ]  
[ STYLE <nStyle> ]  
[ FONT <oFont> ]  
[ TOOLTIP <ctoolt> ]

#### 4.4.9 @ <x>.<y> GET UPDOWN

@ <x>.<y> GET UPDOWN[ <oUpDown> VAR ] <vari>

RANGE <nLower>, <nUpper>  
[ OF <oWnd> ]

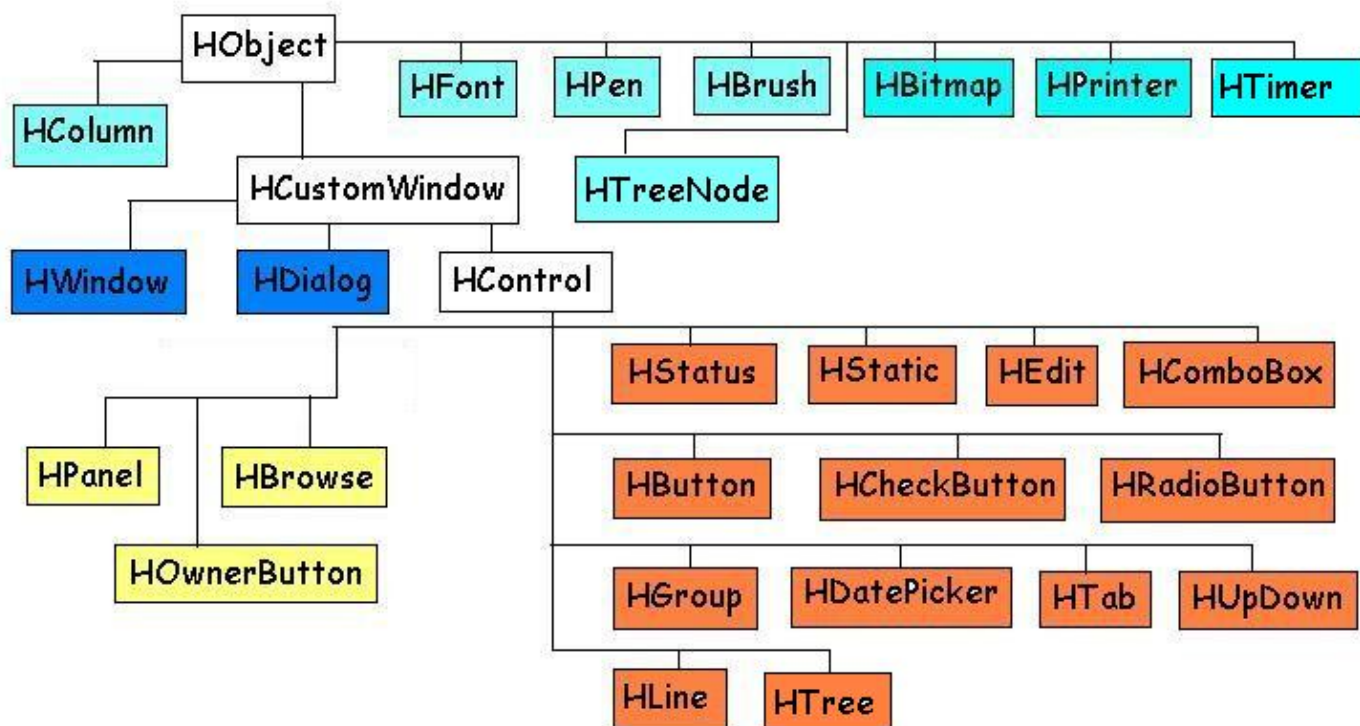
[ ID <nId> ]  
 [ SIZE <width>, <height> ]  
 [ WIDTH <nUpdWidth> ]  
 [ COLOR <tcolor> ]  
 [ BACKCOLOR <bcolor> ]  
 [ PICTURE <cPicture> ]  
 [ WHEN <bWhen> ]  
 [ VALID <bValid> ]  
 [ STYLE <nStyle> ]  
 [ FONT <oFont> ]  
 [ TOOLTIP <ctoolt> ]

## 5 Funções

## 6 Classes

### 6.1 Hierarquia de Classe

O esquema a seguir ilustra a hierarquia de classes, implementada atualmente na HwGUI.



### 6.2 HObject

DATA

classname

### 6.3 HCustomWindow

Não crie instâncias de HCustomWindow. Esta classe serve como base para as classes: janelas, caixas de diálogos e controles. Variáveis e métodos de HCustomWindow provê o comportamento básico que descende por herança da classe, bem como o comportamento que os componentes podem cancelar para personalizar o seu próprio comportamento.

#### DATA

handle  
oParent  
title  
type  
nTop, nLeft, nWidth, nHeight  
tcolor, bcolor, brush  
style  
oFont  
aEvents, aNotify  
aControls  
bInit  
bDestroy  
bSize  
bPaint  
bGetFocus  
bLostFocus  
bOther

#### METHODS

AddControl( oCtrl )  
AddEvent( nEvent,nId,bAction )  
FindControl( nId )

### **6.4 HWindow**

#### DATA

CLASS VAR aWindows  
menu, oPopup

#### METHODS

New( lType,ico,clr,nStyle,x,y,width,height,cTitle,cMenu,nPos,  
bInit,bExit,bSize,bPaint,bGfocus,bLfocus,bOther )  
Activate( nLeft,nTop,nWidth,nHeight )  
AddItem( oWnd )  
DelItem( oWnd )  
FindWindow( hWnd )  
GetMain()  
GetMdiActive()

### **6.5 HDialog**

## DATA

```
CLASS VAR aDialogs
CLASS VAR aModalDialogs
oPopup
lResult
```

## METHODS

```
New( lType,nStyle,cTitle,x,y,width,height,bInit,bExit,bSize,
      bPaint,bGfocus,bLfocus,bOther ) Activate(
nLeft,nTop,nWidth,nHeight )
AddItem( oWnd,lModal )
DelItem( oWnd,lModal )
FindDialog( hWnd )
```

### 6.6 HControl

## DATA

```
id
tooltip
lInit
lHide
```

## METHODS

```
Init()
SetColor( tcolor,bcolor,lRepaint ) NewId() INLINE CONTROL_FIRST_ID
+ Len( ::oParent:aControls )
Hide() INLINE (::lHide:=.T.,HideWindow(::handle))
Show() INLINE (::lHide:=.F.,ShowWindow(::handle))
```

### 6.8 HBrowse

## DATA

active	
winclass	
lChanged	
lDispHead	Devo exibir cabeçalhos?
aColumns	array das HColumn
rowCount	Número de linhas de registros visíveis.
rowPos	Posição da linha atual
colPos	Posição da coluna atual
nColumns	Número de colunas de dados visíveis.
nLeftCol	Leftmost column

xpos	
freeze	Número de colunas a fixar
kolz	Número de registros no browse
tekzp,msrec	
recCurr	
headColor	Cor do texto do cabeçalho
sepColor	Cor dos separadores
tcolorSel,bcolorSel,brushSel	
bSkip,bGoTo,bGoTop,bGoBot,bEof,bBof	
bRcou,bRecno	
bPosChanged	
bEnter, bKeyDown	
internal	
alias	Nome do Alias da base de dados visualizada
x1,y1,x2,y2,width,height	
lEditable	
lAppable	
lAppMode	
lAutoEdit	
lUpdated	
lAppended	

## METHODS

```

New( lType,oWndParent,nId,nStyle,nLeft,nTop,nWidth,nHeight, ;
    bInit,bSize,bPaint,bEnter,bGfocus,bLfocus )
InitBrw()
Rebuild()
Activate()
Redefine( lType,oWnd,nId,bInit,bSize,bDraw,bEnter,bGfocus,bLfocus )
FindBrowse( nId )
AddColumn( oColumn )
InsColumn( oColumn,nPos )
DelColumn( nPos )
Paint()
LineOut()
HeaderOut( hDC )
DoHScroll( wParam )
DoVScroll( wParam )
LineDown()
LineUp()
PageUp()
PageDown()
Bottom()
Top()

```

```

ButtonDown( lParam )
ButtonUp( lParam )
ButtonDbl( lParam )
MouseMove( wParam, lParam )
Edit( wParam,lParam )
Append() INLINE (::Bottom(.F.),::LineDown())
RefreshLine()
Refresh()
ShowSizes()
End()

```

## **6.9 HColumn**

### DATA

```

block,heading,width,type,length,dec,cargo
nJusHead, nJusLin
tcolor,bcolor,brush
lEditable
aList
aBitmaps
bValid,bWhen
bEdit

```

### METHODS

```

New( cHeading,block,type,length,dec,lEditable,nJusHead,nJusLin )

```

## **6.10 HPanel**

### DATA

```

winclass

```

### METHODS

```

New( oWndParent,nId,nStyle,nLeft,nTop,nWidth,nHeight,
      bInit,bSize,bPaint )
Activate()
Redefine( oWnd,nId,nHeight,bInit,bSize,bDraw )
Paint()

```

## **6.11 HOwnerButton**

### DATA

```

CLASSDATA oSelected
winclass
lFlat
state
bClick

```

text,tfont,xt,yt,widtht,heightt  
bitmap,xb,yb,widthb,heightb

## METHODS

New( oWndParent,nId,nStyle,nLeft,nTop,nWidth,nHeight,  
bInit,bSize,bPaint,bClick,lflat,  
aText,aBitmap,cTooltip )  
Activate()  
Redefine( oWndParent,nId,bInit,bSize,bPaint,bClick,lflat,  
aText,aBitmap,cTooltip )  
Paint()  
MouseMove( wParam, lParam )  
MDown()  
MUp()

### **6.12 HStatus**

## DATA

aParts

## METHODS

New( oWndParent,nId,nStyle,aParts,bInit,bSize,bPaint )  
Activate()

### **6.13 HStatic**

## DATA

CLASS VAR winclass

## METHODS

New(  
oWndParent,nId,nStyle,nLeft,nTop,nWidth,nHeight,cCaption,oFont,bInit,bSize,bPaint,ctoolt,tcolor,bcolor  
)  
Activate()  
Redefine( oWndParent,nId,oFont,bInit, ;  
bSize,bPaint,ctoolt,tcolor,bcolor ) SetValue(value) INLINE  
SetDlgItemText( ::oParent:handle,::id,value )

### **6.14 HButton**

## DATA

CLASS VAR winclass

## METHODS



```

New( oWndParent,nId,nStyle,nLeft,nTop,nWidth,nHeight,cCaption,oFont,
      bInit,bSize,bPaint,bClick,ctoolt,tcolor,bcolor )
Activate()
Redefine( oWnd,nId,oFont,bInit,bSize,bDraw,bClick,ctoolt,tcolor,bcolor )

```

### **6.15 HEdit**

#### DATA

```

CLASS VAR winclass
CLASS VAR ICtrl
CLASS VAR IShift
IMultiLine
cType
bSetGet
bValid
cPicFunc, cPicMask
IPicComplex
IFirst
IChanged

```

#### METHODS

```

New(
oWndParent,nId,vari,bSetGet,nStyle,nLeft,nTop,nWidth,nHeight,cCaption,oFont,

bInit,bSize,bPaint,bGfocus,bLfocus,ctoolt,tcolor,bcolor,cPicture,lNoBorder )
Activate()
Redefine(
oWnd,nId,vari,bSetGet,oFont,bInit,bSize,bDraw,bGfocus,bLfocus,ctoolt,tcolor,bcolor,cPicture
)
Init()
SetGet(value) INLINE Eval( ::bSetGet,value )
Refresh()

```

### **6.16 HChecBox**

#### DATA

```

CLASS VAR winclass
bSetGet
value

```

#### METHODS

```

New(
oWndParent,nId,vari,bSetGet,nStyle,nLeft,nTop,nWidth,nHeight,cCaption,oFont,
      bInit,bSize,bPaint,bClick,ctoolt,tcolor,bcolor )
Activate()
Redefine(
oWnd,nId,vari,bSetGet,oFont,bInit,bSize,bDraw,bClick,ctoolt,tcolor,bcolor )

```

Init()

### **6.17 HRadioButton**

DATA

CLASS VAR winclass  
CLASS VAR aGroup

METHODS

New( oWndParent,nId,nStyle,nLeft,nTop,nWidth,nHeight,cCaption,oFont,  
bInit,bSize,bPaint,bClick,ctoolt,tcolor,bcolor )  
Activate()  
Redefine( oWnd,nId,oFont,bInit,bSize,bDraw,bClick,ctoolt,tcolor,bcolor )

### **6.18 HComboBox**

DATA

CLASS VAR winclass  
aItems  
nInit

METHODS

New(  
oWndParent,nId,vari,bSetGet,nStyle,nLeft,nTop,nWidth,nHeight,aItems,  
bInit,bSize,bPaint,bChange,cTooltip )  
Activate()  
Redefine(  
oWnd,nId,vari,bSetGet,aItems,bInit,bSize,bDraw,bChange,cTooltip )  
Init( aCombo, nCurrent )

### **6.19 HGroup**

DATA

CLASS VAR winclass

METHODS

New( oWndParent,nId,nStyle,nLeft,nTop,nWidth,nHeight,cCaption,oFont,  
bInit,bSize,bPaint,tcolor,bcolor )  
Activate()

### **6.20 HFont**

DATA

CLASS VAR aFonts  
handle  
name, width, height ,weight  
charset, italic, Underline, StrikeOut  
nCounter

#### METHODS

Add( fontName, nWidth, nHeight ,fnWeight, fdwCharSet, ;  
fdwItalic, fdwUnderline, fdwStrikeOut, nHandle )  
Select()  
Release()

### **6.21 HPen**

#### DATA

CLASS VAR aPens  
handle  
style, width, color  
nCounter

#### METHODS

Add( style, width, color )  
Release()

### **6.22 HBrush**

#### DATA

CLASS VAR aBrushes  
handle  
color  
nCounter

#### METHODS

Add( color )  
Release()

### **6.23 HBitmap**

#### DATA

CLASS VAR aBitmaps  
handle  
name  
nCounter

#### METHODS

AddResource( name,hDC )  
AddFile( name )  
Release()

#### **6.24 HPrinter**

DATA

hDC

METHODS

New( cPrinter )  
StartDoc()  
EndDoc()  
StartPage()  
EndPage()  
End()  
Box( x1,y1,x2,y2,oPen )  
Line( x1,y1,x2,y2,oPen )  
Say( cString,x1,y1,x2,y2,nOpt,oFont )  
Bitmap( x1,y1,x2,y2,nOpt,hBitmap )

#### **6.25 HDatePicker**

DATA

CLASS VAR winclass  
bSetGet  
value

METHODS

New( oWndParent,nId,vari,bSetGet,nStyle,nLeft,nTop,nWidth,nHeight, ;  
oFont,bInit,bGfocus,bLfocus,ctoolt,tcolor,bcolor )  
Activate()  
Init()

#### **6.26 HUpDown**

DATA

CLASS VAR winclass  
bSetGet  
hUpDown, idUpDown, styleUpDown  
nLower  
nUpper  
nUpDownWidth  
lChanged

METHODS

```

New( oWndParent,nId,vari,bSetGet,nStyle,nLeft,nTop,nWidth,nHeight, ;
oFont,bInit,bSize,bPaint,bGfocus,bLfocus,ctoolt,tcolor,bcolor,nUpDWidth,nLower,nUpper
)
Activate()
Init()
Refresh()

```

### **6.27 HTab**

#### DATA

```

CLASS VAR winclass
aTabs
aPages
bChange,bChange2
oTemp

```

#### METHODS

```

New( oWndParent,nId,nStyle,nLeft,nTop,nWidth,nHeight, ;
oFont,bInit,bSize,bPaint,aTabs )
Activate()
Init()
SetTab( n )
StartPage( cname )
EndPage()
ChangePage( nPage )
HidePage( nPage )
ShowPage( nPage )
GetActivePage( nFirst,nEnd )

```

### **6.28 HTree**

#### DATA

```

CLASS VAR winclass
aItems
himl,aImages,Image1,Image2

```

#### METHODS

```

New( oWndParent,nId,nStyle,nLeft,nTop,nWidth,nHeight,oFont, ;
bInit,bSize,aImages,lResour )
Activate()
Init()
AddNode( cTitle, oPrev, bAction, aImages )
FindChild( handle )
GetSelected()

```

### **6.29 HTreeNode**

## DATA

```
CLASS VAR winclass  
aItems  
handle  
oTree, oParent  
bAction
```

## METHODS

```
New( oTree, oParent, oPrev, cTitle, bAction,aImages )  
AddNode( cTitle, oPrev, bAction, aImages )  
Delete()  
FindChild( handle )  
GetText()
```

### **6.30 HTimer**

## DATA

```
CLASS VAR aTimers  
id  
value  
oParent  
bAction
```

## METHODS

```
New( oParent,id,value,bAction )  
End()
```

### **7 Qhtm – Integração**

#### **7.1 Visão Geral**

QHTM é uma biblioteca em C++, que permite exibir e imprimir conteúdo HTML no seu aplicativo – em qualquer janela, plano de contexto, em um relatório, em um botão ou numa dica de controle. Para mais detalhes veja no endereço: <http://www.gipsysoft.com>.

A HwGUI provê uma interface para esta biblioteca. Para usá-la, você precisa baixar o QHTM do site <http://www.gipsysoft.com/qhtm/freedownload.shtml> e copiar o arquivo qhtm.dll para a mesma pasta onde seu aplicativo está.

**Atenção !!!** O QHTM é liberada sobre licença diferente do Harbour e HwGUI, não esqueça de ler antes de usar!

#### **7.2 Comandos**

```
@ <x>,<y> QHTM [ <oQhtm> ]  
  
[ CAPTION <caption> ]  
[ FILE <fname> ]  
[ RESOURCE <name> ]  
[ OF <oWnd> ];
```

```

[ ID <nId> ]
[ SIZE <width>, <height> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON CLICK <bLink> ]
[ ON SUBMIT <bSubmit> ]
[ STYLE <nStyle> ]

```

Este comando cria o controle QHTM. Html e seu conteúdo pode ser determinado de três maneiras:

- Você pode escrevê-la depois da cláusula CAPTION.
- Pode ser um arquivo .html externo, com o nome informado depois da cláusula FILE.
- Um arquivo de recursos. Neste caso apropriado o nome do arquivo de recurso é informado depois da cláusula RESOURCE.

A cláusula ON CLICK determina o codeblock, que será executado quando o usuário clicar num link externo. A cláusula ON SUBMIT determina o codeblock, que será executado quando o usuário submeter um form.

```

REDEFINE QHTM [ <oQhtm> ]

[ CAPTION <caption> ]
[ FILE <fname> ]
[ RESOURCE <resname> ]
[ OF <oWnd> ];
ID <nId>
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON CLICK <bLink> ]
[ ON SUBMIT <bSubmit> ]

```

Este comand redefine o controle QHTM do arquivo de recursos. O conteúdo Html pode ser informado de três modos e os todos com um comando @ ... QHTM.

```

@ <x>,<y> QHTMBUTTON [ <oButton> CAPTION ] <caption>

[ OF <oWnd> ];
[ ID <nId> ]
[ SIZE <width>, <height> ]
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON CLICK <bClick> ]
[ STYLE <nStyle> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

Este comando trabalha da mesma maneira do @ ... BUTTON, mas o <caption> pode incluir no conteúdo html.

```

REDEFINE QHTMBUTTON <oButton>

```

```

[ OF <oWnd> ]
ID <nId>
[ ON INIT <bInit> ]
[ ON SIZE <bSize> ]
[ ON CLICK <bClick> ]
[ FONT <oFont> ]
[ TOOLTIP <ctoolt> ]

```

Este comando trabalha da mesma maneira do REDEFINE BUTTON, mas o <caption> pode incluir no conteúdo html.

### **7.3 Functions**

QHTM\_Init( [ cDllName ] )

QHTM\_Message( cMessage [,cTitle ] [,nFlags ] )

QHTM\_LoadFile( handle, cFileName )

QHTM\_LoadRes( handle, cResourceName )

QHTM\_AddHtml( handle, cText )

QHTM\_GetTitle( handle )

QHTM\_GetSize( handle )

QHTM\_EnableCooltips()

QHTM\_PrintCreateContext() --> hContext

QHTM\_PrintSetText( hContext,cHtmlText )

QHTM\_PrintSetTextFile( hContext,cFileName )

QHTM\_PrintSetTextResource( hContext,cResourceName )

QHTM\_PrintLayout( hDC,hContext ) --> nNumberOfPages

QHTM\_PrintPage( hDC,hContext,nPage )

QHTM\_PrintDestroyContext( hContext )

### **8 Licença**

A HwGUI é liberada sobre a mesma licença do Compilador Harbour.

*Este programa é um software livre; você pode redistribuí-la e/ou modificá-la sob os termos da GNU General Public License como publicada pela Free Software Foundation; em ambas versões da Licença, ou (na nossa opinião) qualquer versão posterior.*



*This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.*

*You should have received a copy of the GNU General Public License along with this software; see the file COPYING. If not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA (or visit the web site <http://www.gnu.org/>).*

*Como uma exceção especial, você tem permissão adicional para uso do texto contido na liberação da HWGUI.*

*A exceção é que, se você linkar a biblioteca HWGUI com outros arquivos para produzir um executável, isto fará que seu executável não estará coberto pela GNU General Public License. O uso deste executável não está restrito sobre uma conta de código da biblioteca HWGUI inserido e ligado nele..*

## **9 Sobre o Autor**



## **10 Sobre a Tradução**

O objetivo desta tradução é ajudar a comunidade iniciante de desenvolvedores que desejam mais informação em língua nativa. Não é um trabalho perfeito ou acabado, todas as críticas e contribuições serão analisadas com atenção.

Sóstenes

[psostenes@yahoo.com.br](mailto:psostenes@yahoo.com.br)