

# From Clipper to HMG - part 1

This is just a small part of the experience that can be transferred to others who want to use HMG

It is not easy but it's worth

Mail [dragancesu@gmail.com](mailto:dragancesu@gmail.com)

## Content

From Clipper to HMG - part 1 .....	1
Before start.....	2
Clipper.....	3
Harbour .....	3
HMG .....	4
Instalation.....	4
First program.....	5
Compile (from Command Prompt).....	8
Manual .....	9
HMG IDE.....	10
CONTROL Definition .....	10
Samples.....	12
Debug .....	13
First HMG program .....	13
At end.....	16

## ***Before start***

If you used the Clipper then you have many times wanted to write a true Windows program. There are many possibilities for this, but none of them easy.

To be clear at the outset, Windows has its own rules for writing programs that must be respected, that's good, but Clipper is so easy to move is not easy

For starters, you're used to display 80 characters in 24 lines, no more or no less.

Windows defines the different display, the unit of dots on the screen (pixel) so that the program can write / draw on the entire screen.

Let's see a simple Clipper program

```
CLS

n1 = 2
n2 = 2

@ 10,10 GET N1 PICTURE '999'
@ 10,15 SAY '+'
@ 10,20 GET N2 PICTURE '999'
@ 10,25 SAY '='
READ

n3 = n1 + n2
@ 10,30 GET N3 PICTURE '9999'

INKEY(5)

RETURN
```

Compile and have program that adds two numbers. At the beginning clears the screen, define two numbers (N1 and N2), draft a sign of addition and equality. When you start the program enter the first addend and press ENTER, enter the second addend and press ENTER, and a moment later see summation. A little later, the program will be closed.

Note: it is assumed that you know to write and compile clipper program

## Clipper

To recap history. Back in 1980, there appeared a program called dBase III +

<http://en.wikipedia.org/wiki/DBase>

He had a simple interface and solid programming language, supported the DBF format for evolutionary time, the file is stored and data structures.

He was the interpreter, write the command and immediately executed. Of course, it could be written more commands and stored in a file and later performed automatically, such a file had an extension PRG which is known today.

Interpreter is good but the user supplied source code and it did not like developers. Then in 1985 came Clipper, dBase compiler for programs.

[http://en.wikipedia.org/wiki/Clipper\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Clipper_(programming_language))

The result was a stand-alone EXE file that is delivered to customers. Source stayed at programmer, executable version of the user. Great.

The most commonly used versions of the Clipper Summer 87 and 5.01. The latest version of the Clipper 5.3b in 1997

Since the program is written carefully worked in the Windows environment.

There have been many attempts to create a library that will add windows functionality.

## Harbour

Important for us is Harbour project, version 3.0 from the year 2011.

[http://en.wikipedia.org/wiki/Harbour\\_%28software%29](http://en.wikipedia.org/wiki/Harbour_%28software%29)

It was a true compiler for windows clipper and the result was windows 32-bit applications

100% compatible clipper compiler, what needs more? With lots of windows many functions and still is the result of a program in text mode, the screen 80x25 lines and the problem was printed on the Windows-only printers

Not long awaited solution to these problems

<https://harbour.github.io/third-party.html>

There are commercial solutions but also free which are interesting for us. There are also solutions for Linux

## **HMG**

One good solution to (almost) all can leave to move on to the windows HMG

<https://sites.google.com/site/hmgweb/>

No matter how good the program, support is welcome, HMG has an excellent forum that accompanies many people from all over the world

<http://www.hmgforum.com/>

There you will find the latest version of the program

<http://www.hmgforum.com/site/>

Now the current version 3.3.1 you need to download and install. At the end of installation, set the system variable, add C:\HMG in the PATH

Note: The program is installed in c:\hmg.3.3.1 but will be written to c:\hmg

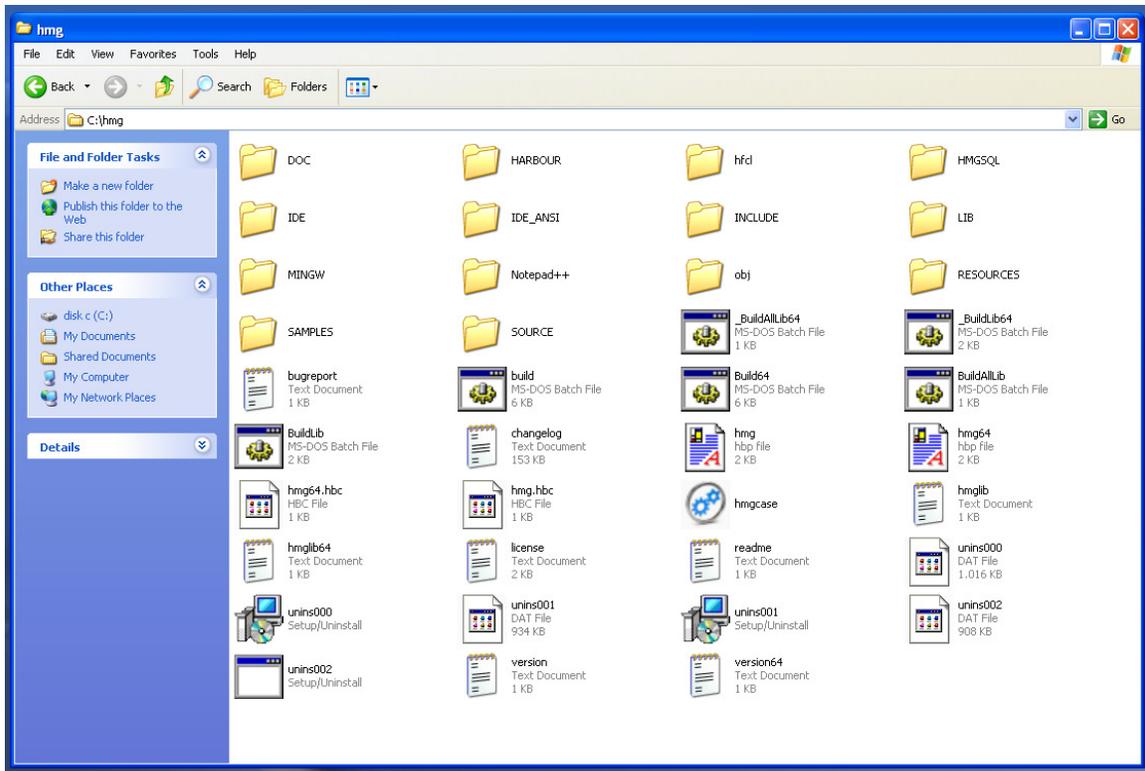
## ***Instalation***

At the end of the installation you will have a folder C:\HMG and its a lot of folders,

DOC – Documentation  
HARBOUR - harbour compiler  
MINGW - MiniGW compiler  
Notepad ++ - excellent editor  
IDE - GUI interface

For the compilation of the program should be run BUILD.BAT

New extensions are  
HBP – harbour project  
RC - resource file

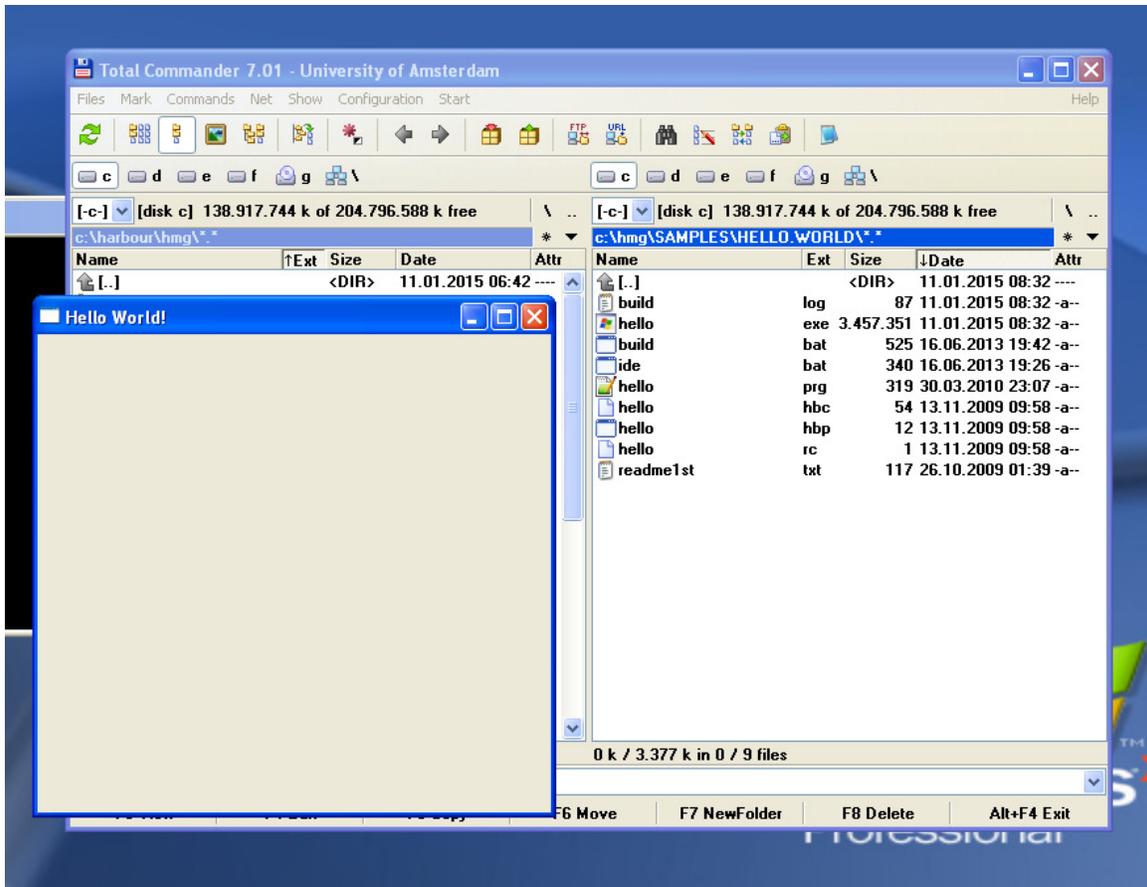


## First program

Name	Ext	Size	Date	Attr
[.]	<DIR>		11.12.2013 22:57	
build	bat	525	16.06.2013 19:42	-a-
ide	bat	340	16.06.2013 19:26	-a-
hello	prg	319	30.03.2010 23:07	-a-
hello	hbc	54	13.11.2009 09:58	-a-
hello	hbp	12	13.11.2009 09:58	-a-
hello	rc	1	13.11.2009 09:58	-a-
readme1st	txt	117	26.10.2009 01:39	-a-

A classic example in this for many programs is HelloWorld, which you will find in the folder SAMPLES

There you have it all and just need to type Build hello.prg



The result is immediately visible, set in the middle of the screen

To see the program

```
/*  
* HMG Hello World Demo  
* (c) 2002-2009 Roberto Lopez <mail.box.hmg@gmail.com>  
*/
```

```
#include "hmg.ch"
```

```
Function Main
```

```
    DEFINE WINDOW Win_1 ;  
        ROW 0 ;  
        COL 0 ;  
        WIDTH 400 ;  
        HEIGHT 400 ;  
        TITLE 'Hello World!' ;  
        WINDOWTYPE MAIN
```

```
END WINDOW

Win_1.Center

Win_1.Activate

Return
```

Let's analyze it

*/\* Text \*/ - this is how you write a comment*  
*// Text - and that's the way till end of row*

**#include "hmg.ch" - necessary include files, there is a defined everything you need**

**Function Main - at the beginning should be defined as the function / procedure calls**

```
DEFINE WINDOW Win_1;
  ROW 0;
  COL 0;
  WIDTH 400;
  HEIGHT 400;
  TITLE 'Hello World!' ;
  WINDOWTYPE MAIN
```

**// Here to write controls**

**END WINDOW - this is part without which it can not, must define the main window, the name (Win\_1), size (WIDTH 400 HEIGHT 400), title (TITLE 'Hello World! ') and type (WINDOWTYPE MAIN)**

**// Win\_1.Center - this is the possibility that the program displays centered on the screen, not necessarily**

**Win\_1.Activate - this is run on-screen display**

**Return - standard end of program**

## **Compile (from Command Prompt)**

BUILD.BAT procedure used for the compilation of programs

Usage:

- (1) build program.prg
- (2) build project.hbp

With Clipper've had RMAKE program that was used for (incremental) compilation project

There is no similar program but there HBP file where you list all the modules that should be compiled and linked into the application (EXE file)

# Manual

HMG is a lot added clipper program and initially it is not easy, but the authors did an excellent reference guide, which you will find in C:\HMG\DOC\hmgdoc.htm

HMG Reference (c)2002-2014 Roberto Lopez <http://sites.google.com/site/hmgweb> - Windows Internet Explorer

C:\hmg\DOC\data\index.htm

File Edit View Favorites Tools Help

★ Favorites ☆ Suggested Sites Free Hotmail Web Slice Gallery Like Music - Try AOL!

HMG Reference (c)2002-2014 Roberto Lopez <http://...

- Basics
  - Building HMG Applications
  - Tutorial (English)
  - Tutorial (Español)
  - FAQ
- HMG IDE
- Windows
- Controls
- Properties
- Events
- Methods
- Functions
- Misc. Commands
- Print System
- Report Writer
- HMG HPDF
- Bos Taurus
- Advanced
- External Guides and Tutorials
- HFCL
- HMG UNICODE
- HMG 64-bits

## HMG TUTORIAL

Sample Files: \hmg\samples\tutorial

### Your First HMG Program

I'll not be original, so this program will display a 'Hello World' message :)

```
#include "hmg.ch"

Function Main

    DEFINE WINDOW Win_1 ;
        AT 0,0 ;
        WIDTH 400 ;
        HEIGHT 200 ;
        TITLE 'Tutor 01 - Hello World!' ;
        MAIN

    END WINDOW

    ACTIVATE WINDOW Win_1

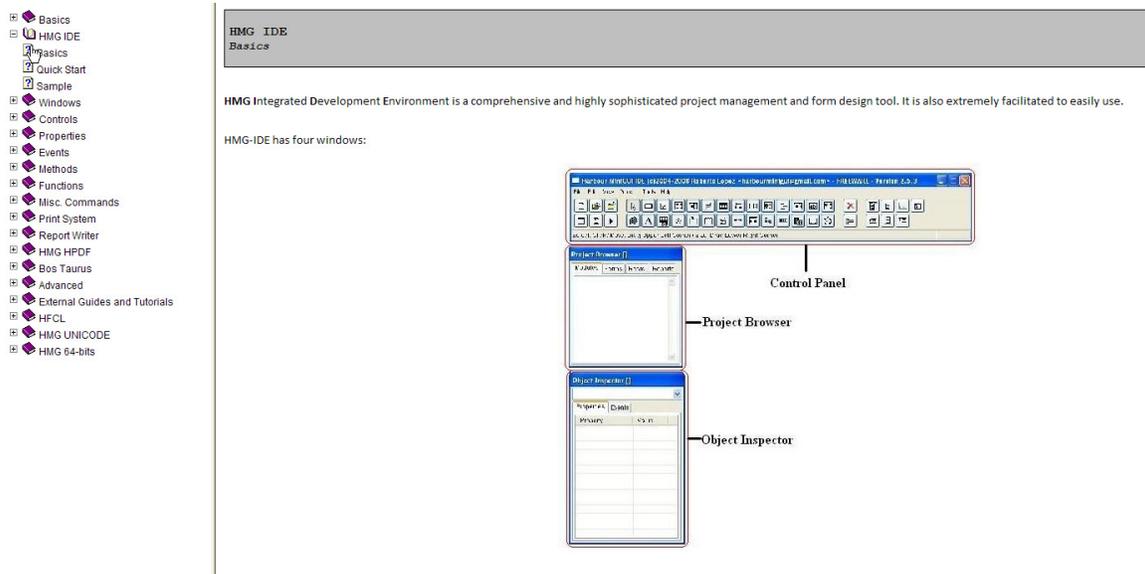
Return
```

- DEFINE WINDOW command: Will create the main window for the program.
- Win\_1: Is the name of the window.
- AT 0,0: Indicates the window position (row=0,col=0)
- WIDTH 400: Means that the window will have 400 pixels width.
- HEIGHT 200: Means that the window will have 200 pixels height.
- TITLE 'Hello World!': Indicates the text in the window title bar.
- MAIN: Indicates that we are defining the main application window (a main window is required for all HMG applications)
- ACTIVATE WINDOW Form\_1: Will show the window and start the event loop.

## HMG IDE

The authors have made an effort to make a program that will help beginners to the HMG IDE that will help to define the window and set the control to it

The result is a (project) HBP, (the program) PRG and (form) FRM.



User HMG IDE is complete so you can read it and try to make a simple program

The tool is excellent but maybe confusing for a beginner, I rarely use it

## CONTROL Definition

There you have all the controls that are needed for the program, and can be defined in two ways:

Do not be surprised if in a program, see the definition and the one and the other way, that HMG supports only a matter of taste what you use

- Windows
- Controls
  - This Object
  - Activex
  - AnimateBox
  - Browse
  - Button
  - CheckBox
  - CheckBoxButton
  - ComboBox
  - Context Menu
  - Control Context Menu
  - DatePicker
  - DropDown Menu
  - EditBox
  - Frame
  - Grid
  - HyperLink
  - Image
  - IpAddress
  - Label
  - ListBox
  - Main Menu
  - MonthCalendar
  - Notify Menu
  - Player
  - ProgressBar
  - RadioGroup
  - RichEditBox
  - Slider
  - Spinner
  - SplitBox
  - StatusBar
  - Tab
  - TextBox
  - Timer
  - TimePicker
  - Tree
  - ToolBar

**@...FRAME / DEFINE FRAME**  
*Creates a frame control*

**Standard Syntax (xBase Style):**

```
@ <nRow> ,<nCol>
  FRAME<ControlName>
  [ OF | PARENT <ParentWindowName> ]
  [ CAPTION <cCaption> ]
  WIDTH <nWidth>
  HEIGHT <nHeight>
  [ FONT <cFontName> ]
  [ SIZE <nFontSize> ]
  [ BOLD ]
  [ ITALIC ]
  [ UNDERLINE ]
  [ STRIKEOUT ]
  [ BACKCOLOR <aBackColor> ]
  [ FONTCOLOR <aFontColor> ]
  [ TRANSPARENT ]
```

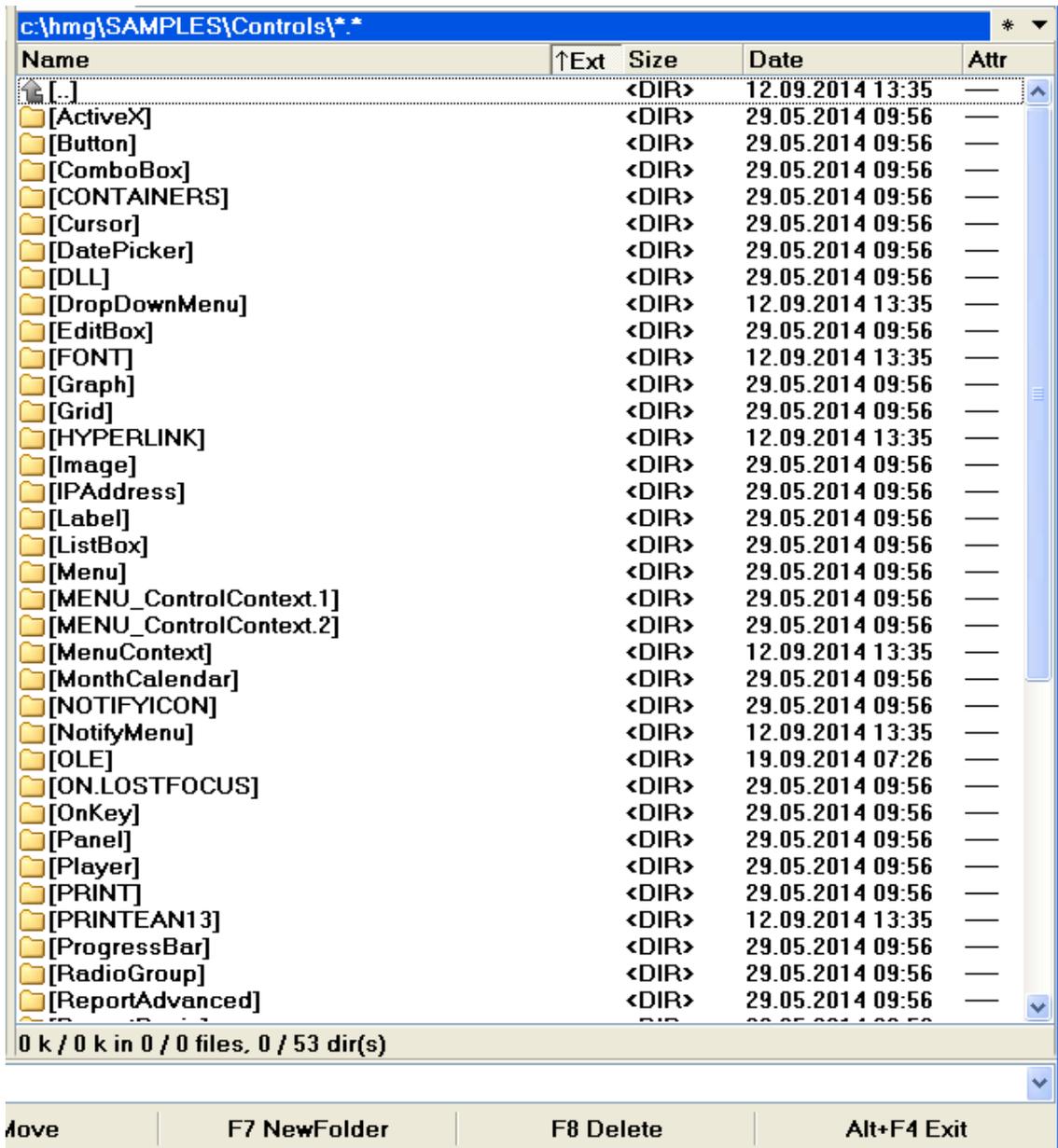
**Alternate Syntax:**

```
DEFINE FRAME <ControlName>
  PARENT <ParentWindowName>
  ROW <nValue>
  COL <nValue>
  WIDTH <nValue>
  HEIGHT <nValue>
  FONTNAME <cValue>
  FONTSIZE <nValue>
  FONTBOLD <lValue>
  FONTITALIC <lValue>
  FONTUNDERLINE <lValue>
  FONTSTRIKEOUT <lValue>
  CAPTION <cCaption>
  BACKCOLOR <aBackColor>
  FONTCOLOR <aFontColor>
  TRANSPARENT <lValue>
END FRAME
```

## Samples

Examples for all controls will found in the folder c:\hmg\samples\controls

Try it and see how they look and how they work



## Debug

Clipper has an excellent debugger, and it exists in HMG

In order to run the program, it is necessary to call a function Altd (), a compilation do with the BUILD / b program.prg

It's not perfect, and many times we did not work

## First HMG program

At the beginning, we had an example Clipper program and gather two numbers. This is the version of the program HMG

You will notice that defined the MAIN WINDOW, then three TEXTBOX (input fields), one LABEL (text on screen) and one BUTTON (button)

```
#include "hmg.ch"

Function Main

// set navigation extended

    DEFINE WINDOW Form_1 ;
        AT 0,0 ;
        WIDTH 640 HEIGHT 480 ;
        TITLE 'HMG Demo - Add 2 Number' ;
        MAIN

        // ON KEY ESCAPE ACTION Form_1.Release

        @ 100,100 TEXTBOX Num_1 ;
            VALUE 2 ;
            NUMERIC ;
            MAXLENGTH 5 ;
            RIGHTALIGN ;
            WIDTH 20

        @ 105,130 LABEL Plus ;
            VALUE ' + ' ;

        @ 100,150 TEXTBOX Num_2 ;
```

```
VALUE 2 ;  
NUMERIC ;  
MAXLENGTH 5 ;  
RIGHTALIGN ;  
WIDTH 20
```

```
@ 100,200  BUTTON ADD_ING ;  
CAPTION ' = ' ;  
ACTION add_two_number () ;  
WIDTH 30 ;  
HEIGHT 25
```

```
@ 100,250 TEXTBOX Num_3 ;  
VALUE 2 ;  
NUMERIC ;  
MAXLENGTH 5 ;  
RIGHTALIGN ;  
WIDTH 30
```

```
END WINDOW
```

```
Form_1.Center
```

```
Form_1.Activate
```

```
Return Nil
```

```
function add_two_number ()
```

```
Form_1.Num_3.Value := Form_1.Num_1.Value +  
Form_1.Num_2.Value
```

```
Return
```

For the compilation type BUILD prog\_1h.prg and you will see the result

Unlike Clipper to Windows applications to move to the next field is performed by pressing the Tab

To have acted as a Clipper in order to remove a comment SET NAVIGATION EXTENDED and then pressing the Enter move to the next field

The Clipper is usually used

```
READ
```

```
IF LASTKEY () = 27 // ESC
```

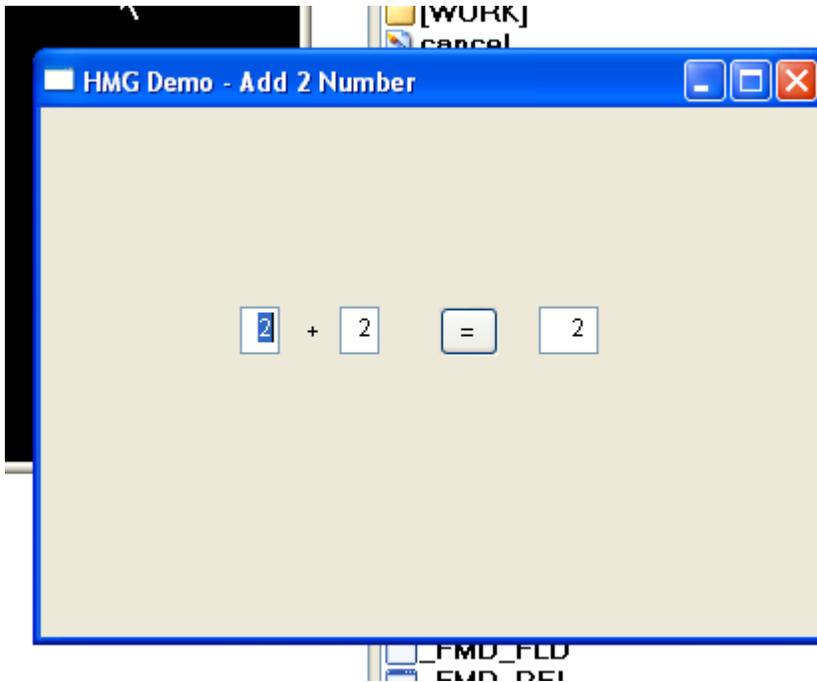
RETURN

...

Here there is not and to exit the program should click on the red X in the upper right part of the window

This is the default behavior in Windows environment, but there is a possibility of defining actions for key (similar to a button), and if you remove the comment to the line `ESCAPE KEY ACTION Form_1.Release` Pressing the ESC program will be completed.

Note: be careful with this because this closes the whole program and not break any action, it should be defined differently



## ***At end***

For now, this

HMG is in any case a great program that will allow you to create Windows programs you want them

It is not easy, but it is not difficult. You just need to accept a new concept and that's all

With HMG and installation and you have Notepad ++ programming editor will facilitate your everyday work

Unfortunately, it is not enough just to compile the old applications, working DOS and WINDOWS program is different and programs should be written anew

I think it's worth