



ViziBasic v2.2

July 28, 2009

Index

What is ViziBasic?	2
Quick introduction to iziBasic	3
An example of designing a form with iziBasic and ViziBasic	4
Contact Information	5
iziBasic users group & resources	6
How to install ViziBasic?	6
How to use ViziBasic?	7
Using the menu	7
How to select an object or a graphic on the form to update it	10
Objects and Graphics IDs in ViziBasic	11
Tutorial - my first ViziBasic program	12
Appendix #1: Sample ViziBasic project files	15
Appendix #2: ViziBasic versions history / log	17





What is ViziBasic?

Notice: in French "izi" is pronounced like "easy" in English

ViziBasic stands for Visual easy Basic for Palm. It is an add-on application that I especially made for easing the development of iziBasic projects requiring GUI components (well, most of the Palm applications that you will develop should need that!).

ViziBasic is a Visual editor for iziBasic.

ViziBasic is an application dedicated to:

- design simple GUI applications, with forms and objects in a form
- have objects perform specific actions thanks to attached iziBasic source code to each object, source code that you write to customize these actions
- generate and write the iziBasic source code to some Palm DOC format file that can then be edited like any other iziBasic source code for further enhancing your application
- thanks to its integration with the iziBasic compiler (as its name states it!), compile directly the generated iziBasic source code for immediate result

And all of this is done directly on board of your Palm device.

So, ViziBasic is a Rapid Application Development tool (so called RAD), a "What You See Is What You Get" (WYSIWYG) form designer, and the perfect companion to the iziBasic compiler.

ViziBasic is then especially useful for quickly prototyping an application and, since the source code is written in the standard Palm DOC format, it can be the base for further enhancements or more complex coding.

ViziBasic does not come with iziBasic and it is quite useless without it.

You have to purchase the iziBasic compiler separately (\$25).

ViziBasic requires a full version of iziBasic to compile the source codes it generates.

But, in the case you do not own a full version of iziBasic, you may still play around with its user interface and generate the related source codes to help you make up your mind before purchasing it.

Last but not least, ViziBasic itself is coded with iziBasic ☺



Quick introduction to iziBasic

iziBasic is a high level development compiler which builds Stand-alone applications. The great thing is that it does all of that directly on-board of your Palm OS based device. This Stand-alone application builder is very convenient for those wishing to distribute their creations made with iziBasic in a ready to run software.

Source codes are easily written using:

- either the Memo Pad application which is shipped with all Palm OS devices,
- or an on-board third party DOC editor of your choice.

iziBasic reads the source codes, compiles them and builds the applications. The use of the Memo Pad is very convenient as you do not need to install any additional editor software to iziBasic. The 4096 characters limit of a memo is not an issue as iziBasic is able to link (or “chain”) together up to 10 memos to build a bigger application. But, for your convenience, you may as well use a DOC editor which overpasses this 4096 characters limit.

As its name also states it, iziBasic uses the BASIC high level and very easy to learn development language, a customized subset of it to be precise. You will discover how easy and quick it is to develop software with iziBasic when the more common development tools available on the Palm OS platform usually require pretty good development skills.

Easiness of the Basic language and Palm hosting are not gained against execution speed of compiled programs. According to my Bench2 analysis (<http://www.aldweb.com/articles.php?lng=en&pg=24>), iziBasic is one of the Palm hosted tools which generate the fastest programs. It is in 6th position, behind the PP, OnBoardC and QuartusForth (which are compilers), PLua and my LaFac-HELP (which are runtime based like iziBasic). And iziBasic is the fastest on board Basic dialect.

An example of designing a form with iziBasic and ViziBasic

In the following example, we design a simple form with the classical "Hello World" message stored in a label.

In the standard iziBasic use, you input all of your source code manually:



We write all of our source code, following iziBasic's syntax

Compiling this source code is required to see the result

Were we to want to change the label's position, we would need to go back in the source code, then compile again...

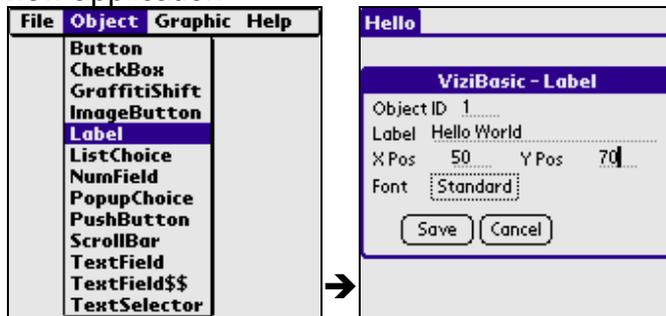
With ViziBasic, you may do the same, but in a visual way:



Result is immediately shown on the screen (title)



First, we define the main characteristics of our new application



Result is again immediately shown on the screen



Then, we add a label and a wizard window helps us define all required characteristics for such an object

To change the label's position, we just need to select it to go back to the previous window



Contact Information

- ✓ World Wide Web main download Site : <http://www.aldweb.com>
- ✓ Author e-mail : info@aldweb.com

ViziBasic is a shareware.

ViziBasic does not come with iziBasic and it is quite useless without it. You have to purchase the iziBasic compiler separately (\$25).

The limitations of the trial version are very light, so that you can really play with it and decide if it is worth for you. Therefore, all functionalities are activated.

The limitations are:

- A nag screen to remind you to buy the full version
- You may use it during 2 weeks, then you have to buy it to keep using it

To get a full version of ViziBasic, please refer to the ViziBasic.txt file that was shipped together with this software package or look for ViziBasic on my web site (<http://www.aldweb.com>) and follow instructions.

The cost of ViziBasic is just as little as \$15.

When you register, you receive a full version of ViziBasic that you just need to install on top of the trial version.

Thanks for purchasing ViziBasic.





iziBasic users group & resources

Upon the request of many iziBasic early adopters, I have set up a forum on my web site (<http://www.aldweb.com>) to give you an opportunity to exchange tricks, ask for help & support and share whatever else you would like with the growing community of iziBasic developers.

I will be on the iziBasic forum very often too. So, as to have most people benefit from good ideas, upgrades features requests, beta versions... please use this iziBasic sharing forum rather than sending me e-mails. Then, everybody will get the benefit of your input ☺

Other resources have been created to help you with the iziBasic use: a Yahoo! User Group, a Wiki (both thanks to Montalvo from Mexico), explanations and samples about the so called "PP applets" (thanks to Khertan from France). All links to these resources can be found on this web page:

<http://www.aldweb.com/articles.php?lng=en&pg=6085>.

How to install ViziBasic?

There is nothing special to say here.

ViziBasic is a PRC file that is installed like any other Palm file using HotSync.

So, extract **ViziBasic_trial.PRC** (or **ViziBasic_full.PRC** if you bought the full version) from the ZIP archive file.

Double-click on it and the Palm install tool will popup.

ViziBasic.PRC will be transferred to your Palm device next time you synchronize your Palm with your PC with HotSync.



Please uninstall any previously installed version of ViziBasic before installing this one.

- ✓ **Minimum Palm OS requirement for ViziBasic is version 3.5 with 4 MB of RAM**
- ✓ **ViziBasic is Palm OS version 5 compliant**

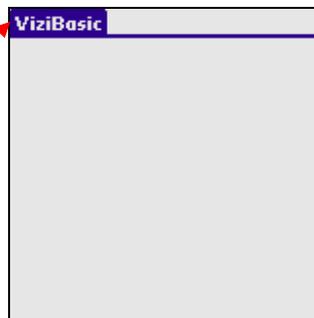
How to use ViziBasic?

I'll be quick in these explanations as ViziBasic behaves like all Palm OS based applications, so it is very intuitive to use. ☺

Using the menu

As the objective of ViziBasic is to design a user interface, the whole screen is dedicated to this purpose and no predefined object can be put on the screen. Therefore, all of the building of this user interface is accessed through the menu.

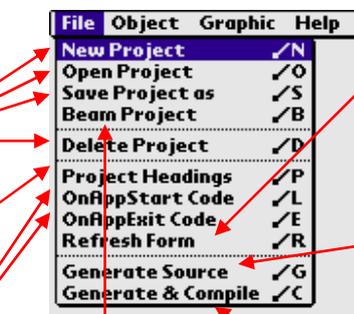
Accessing the menu is done like with any other Palm application, either by clicking on the form's title or by clicking on the menu button 



The File menu offers all standard functions to manage your project files (New, Open, Save, Delete)

Project Headings will let you modify the main characteristics of your project

OnAppStart Code and OnAppExit Code let you write some code to be executed either at program start or when the application exits



Beam Project lets you beam the project file to another device

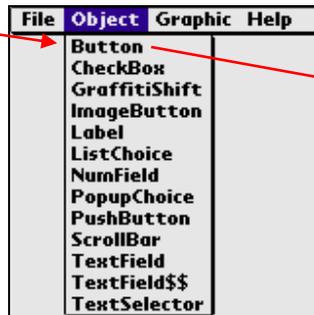
Refresh Form will erase the screen and draw back all objects and graphics in the given order of their IDs (see below)

Generate Source will save your project in a standard Palm DOC format that can then be the base for further development

Generate & Compile will generate the DOC file and launch iziBasic for compiling the project

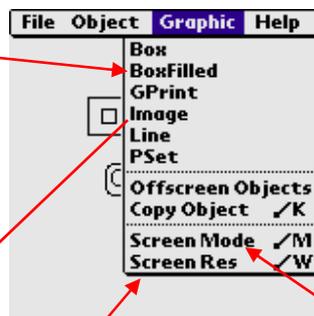
Note: most of these functions will be presented in depth in the Tutorial - my first ViziBasic program

The Object menu lets you insert all possible objects in your form (please refer to the iziBasic user manual, GUI section, to get detailed explanations about these objects)



For instance, you can define all parameters for a button

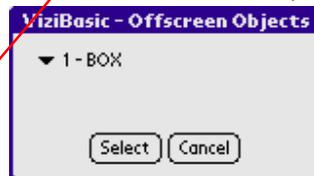
The Graphic menu lets you insert all kinds of graphics (please refer to the iziBasic user manual, Graphics section, to get detailed explanations about these graphics)



The Offscreen Objects option lets you select objects and graphics that you would have defined out of the bounds of the screen and that you can therefore not select with the stylus (see next page)



For instance, you can define all parameters for an image



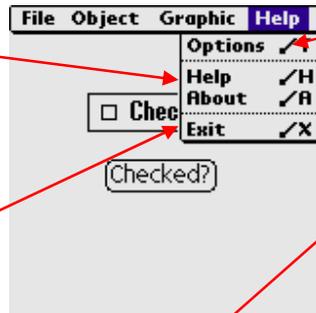
It is also in this menu that you can change the screen mode on the fly to see the resulting of your user interface on different devices that your software could be used on



Screen resolution can be activated on Palm devices with 480x320 high resolution screens (Palm T3, T5, TX)



The Help menu is very common too, offering access to a quick Help page and the About window

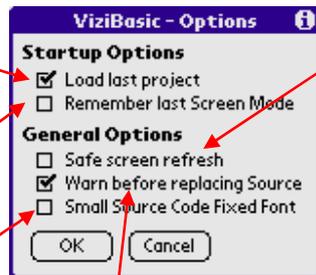


The Options lets you customize some parameters for the ViziBasic interface

Exit closes ViziBasic and returns to the launcher (quite obvious!)

If checked, the last project will be opened when ViziBasic is started

If checked, ViziBasic will be opened with the last black&white, 4 grays, 16 grays or 256 colors screen resolution. Otherwise, the best screen resolution available on the device will be chosen.



If checked, ViziBasic will popup a warning message in the case you are about to generate an iziBasic Source code file and this file already exists. You will then be given the choice of confirming or not the replacement of the previous Source code.

When unchecked, ViziBasic will sometimes crash when refreshing display. This seems to occur on older devices (Palm OS < 5) when many objects are drawn on the screen. This problem is related to a Palm OS memory releasing issue.

Check this option (refresh will be a little bit slower) if you ever suffer from such a crash leading to a soft reset. Otherwise, for performance reasons, you better leave it unchecked.

This option is shown only if your device supports high resolution screen (320x320 or 320x480 pixels). You may decide what size the source code font should be. Be aware that Portrait and Landscape Screen Resolutions are not managed by ViziBasic when generating the iziBasic source code.

Your code should manage them, for example by checking that the target device accepts such a Screen Resolution by using the HIGHRES(2) function and also by using the SETRES statement, most probably in the OnAppStart Code routine.

Also, depending on your SETRES settings, your code should potentially manage the display changes when the 1003 value is returned by the DOEVENTS and WAITEVENT functions. This can only be done in your favorite source code editor, ViziBasic does not provide any tool for this purpose.

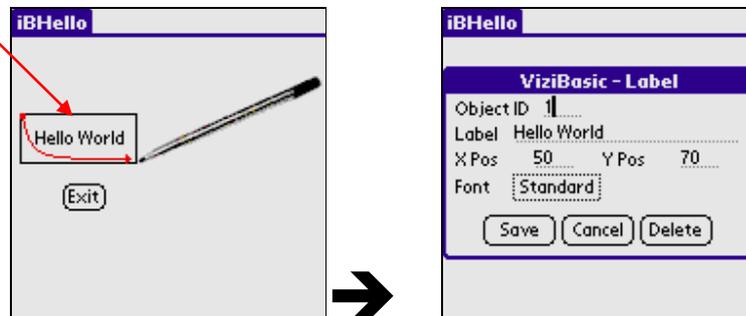
How to select an object or a graphic on the form to update it

There are 2 options to select an object or a graphic on the form in order to then change its characteristics:

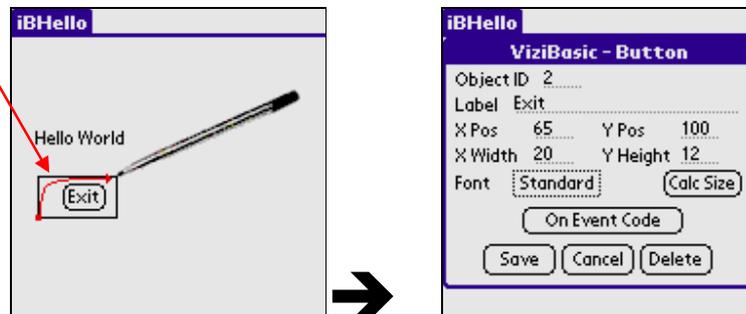
1. If the “thing to select” is a graphic, a label, a numbox or a textbox, you have to surround it by a square which is drawn when you put the stylus on the screen and drag it somewhere else.
2. If the “thing” to select is an object that answers to an event (like a button or a checkbox), you can either tap it with the stylus or proceed the same way as you do for a graphic, meaning drawing a square around it.

Let's see it with screenshots:

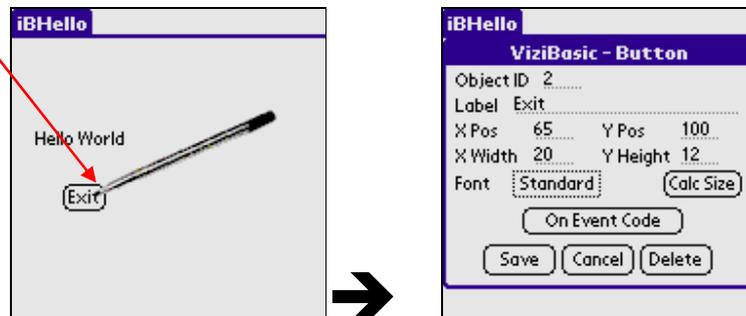
In this case, we decided to select a label, so we put the stylus on one corner of the label (top left in the screenshot) and move it while keeping it on the screen to the opposite corner (bottom right)



Now we decided to select the button the same way as previously for the label, but starting at the bottom left part and up to the top right part



But we could have selected the button by just clicking on it once
 Note: this way of doing is only possible for objects which answer to the click event and it is therefore not possible for graphics



Note: see the Offscreen Objects option in previous page for objects that you cannot select the standard way because you defined them out of the bounds of the screen.



Objects and Graphics IDs in ViziBasic

Objects, just like in iziBasic, require a unique ID to be referenced and then used. But, graphics do not have an ID in iziBasic as they are just drawn on the screen and then “forgotten”.

In ViziBasic, both objects and graphics are referenced with a unique ID, which is the key used by ViziBasic to remember them. The ID will be transferred to the iziBasic source code for the objects. The ID has to be in the [1..899] range as IDs in the [900..999] range are reserved for ViziBasic’s use.

This ID is also used by ViziBasic to know in which order to draw and display both graphics and objects. Object or graphic with ID 1 is drawn before object or graphic with ID 2, and so on.

This is especially useful, for instance, when you want to draw a graphic or an object over another one. The order in which they are drawn is then very important. You would not want a button being hidden by a filled rectangle around it...

An ID is automatically proposed when a new object or graph is drawn, taking the biggest known ID so far and it is incremented by one unit.

ViziBasic only refreshed the current object or graphic when you have finished updating its characteristics. But when you use the Refresh option available in the File menu (see above), all objects and graphics are sorted back and drawn in the right order.

This is the end of this quick tour. ViziBasic is just that easy to use. 😊

The tutorial below will give you a first practice.

Tutorial – my first ViziBasic program

In this tutorial, we will design a very simple application made of 2 objects and one graphic.

One of the two objects will be a checkbox that the user can check or uncheck.

We will put our checkbox in a frame... yep... let's say, for cosmetic reasons!

The other object will be a button, which, when pressed, will popup a message telling if the checkbox is currently checked or not.

Our first step will be to create a new project. For doing so, let's click on the menu button (or on the title bar which displays "ViziBasic" the first time), open the File menu and choose the New Project item. The window on the right side will pop up.

When you are finished filling in the form, don't save your changes yet, click on the AboutBox button instead

Don't forget to change your changes at this stage, and again when you find yourself back on the previous window

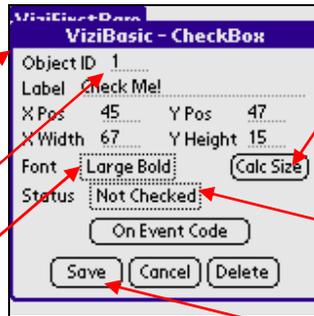
Mandatory fields are the application name and the Creator ID for the application (please refer to the iziBasic documentation for the Creator ID importance in Palm OS programming). All other parameters are facultative ones and are linked to their related Compiling Directive in iziBasic.

The AboutBox will let us personalize the About box of our application.

We could also include some general source code in the Headings Code, that could be called from various objects in our application. But, we will not do so in our first basic application.

Now, let's insert our checkbox. For doing so, go in the Objects menu and select the CheckBox item.

ViziBasic proposes 1 as Object ID (first object). The font could easily be changed from Standard (default value) to Large Bold



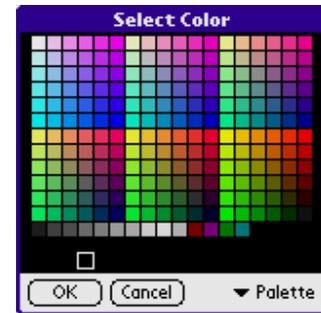
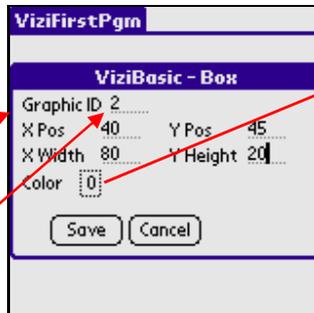
The Calc Size button automatically calculates the smallest X Width and Y Height coordinates for the given object and, important, the given font.

We will leave the initial status as unchecked.

We create the checkbox by pressing the Save button.

Now, we move on to adding a frame around the checkbox. For doing so, we go to the Graphics menu and choose the Box item.

ViziBasic proposes 2 as Graphic ID (second object).

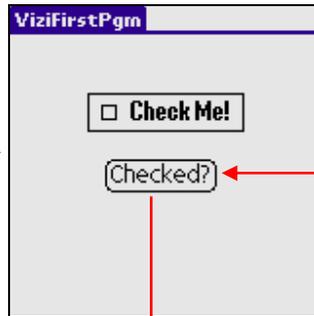


If wanted, the drawing color can be changed.

And, finally, let's add a button. Now, you have guessed that you should go in the Objects menu and choose the Button item.



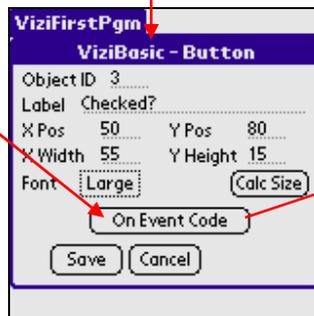
If everything went fine for you so far, you should find yourself with something very similar to, not to say exactly as, the screenshot on the right side.



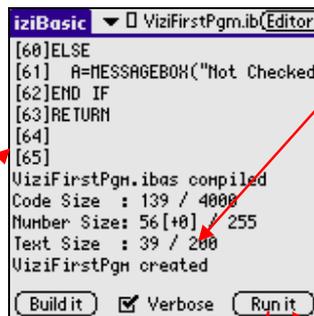
Our user interface is ready. Now we have to make the application do something!

So will click on the button to put some iziBasic code being the result that the user of the application will expect when he will press the button.

Here, we will press the On Event Code button to have some customized iziBasic code runned when the user will press the button



We are all set now. We just have to compile this application with iziBasic. So, we go in the File menu, and choose the Compile item. This launches iziBasic and compilation runs.



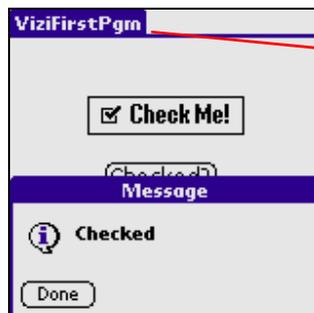
If you made no mistake in the above source code, your application should be created.

Note: the source codes generated by ViziBasic require the full version of iziBasic to be compiled; so the trial version will not compile them.

Or we may choose to only Generate the Source and can then browse it with any DOC editor (PIAF in the screenshot)



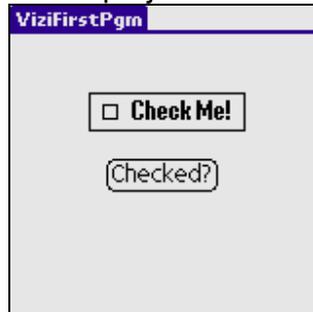
We may now run it.



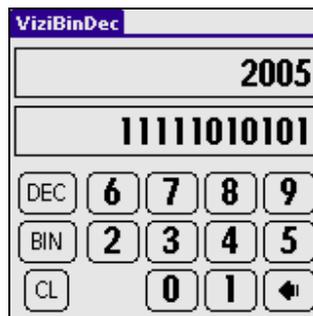
Appendix #1: Sample ViziBasic project files

In the ViziBasic ZIP distribution file, you will find four sample project files:

- ✓ ViB_ViziFirstPgm.pdb – this is the project file of the above tutorial



- ✓ ViB_ViziBinDec.pdb – this is a simple Binary to Decimal and Decimal to Binary convertor



- ✓ ViB_DayOfWeek.pdb – this is a simple day of week finder for any date



- ✓ ViB_ViziStudio.pdb – this is a first level of development studio



Reading and working on these project files will help you to quickly get a full understanding of the ViziBasic capabilities as I believe that learning by the example is the best way to learn programming.

Therefore, I highly invite you to give a deep reading to project files and their iziBasic source codes, and to run these sample programs.

Note: as the source codes generated by ViziBasic require the full iziBasic to be compiled; the trial version will not compile them. But, I provide a compiled and running version for these sample programs:

- ✓ *ViziFirstPgm.prc*
- ✓ *ViziBinDec.prc*
- ✓ *DayOfWeek.prc*
- ✓ *ViziStudio.prc*



Appendix #2: ViziBasic Versions history / log

Next versions roadmap

- *Display Image from Resource File if defined*
- *Drag & drop objects and graphics to adjust their position in the form in a more user friendly manner*
- *Implement any good idea that you would have ☺ ...*

v2.2 (07/28/09)

- bug fix: Edit menu now effective in the "Save Project as" window
- renamed "Screen Resolution" to "Screen Mode" in the Options screen, as this wording is the appropriate one for what it does
- changed Screen Mode selection from 4 menu options to one dedicated selection form
- added a confirm deletion of source code notice to avoid deletion by mistake!
- added a new "Beam Project" menu item to let you beam project files between devices
- added scrollbars to source code editing fields, for easier navigation
- source code editing fields now have a fixed width font, for easier coding
- added option to select size of this fixed font (small/highRes or big/lowRes)
- added option to create a new object as a copy from an existing object (including related event's source code)
- added a new menu option to manage HighRes+ mode available on Palm T3,T5 and TX devices
- added display of the "A\$(n)" case for PopupChoice and ListChoice objects
- added two new sample programs: DayOfWeek and ViziStudio

v2.1 (09/17/07)

- recompiled this program with the latest iziBasic version 6.1 to avoid device freezes or soft resets on some devices



v2.0 (09/20/06)

- when loading last project at program launch, now checks if the project file was not deleted in the meantime
- added an option to select an item by ID for objects or graphics drawn out of screen's bounds
- Heading Code is now prefilled with "_GosubLabel: / RETURN" if empty
- added OnAppStart and OnAppExit source code procedures
- added an option to warn and get user validation before replacing Source Code DOC file
- auto migrates project files of previous versions to new format
- upgraded the ViziBinDec sample application to show the "0" initial value, using the new OnAppStart source code procedure, and added an icon resource to this sample application
- added some explanations in the user manual of how to browse in Code and TEXTFIELD\$\$ screens without scrollbar
- performed massive optimizations in the source code

v1.1 (11/14/05)

- bug fix: the SCROLLBAR object is now generated in the source code generation process

v1.0 (11/04/05)

- initial release for Palm OS