



VBMax Liquid Crystal Display DLL

Copyright © 1995 - 1996 Mike Stanley. All Rights Reserved.

VBMax—Visual Basic to the Max

Table of Contents

- Description
- Registration
- Important—Use at Your Own Risk!
- Installation
- Frequently Asked Questions
- Technical Support
- Gronked Files
- Packing List
- How to use the DLL
- Methods Summary
- Properties Summary
- Notes on Color Selection

Description

A fully functional in-process OLE server for adding LCD style controls to your VB 4.0 applications.

VBMaxLCD.dll contains methods and properties for setting the digit and background colors, autosizing the display, blinking colons, flashing digits, showing or hiding unlit segments and aligning digits left, right or centered.

Written entirely in VB 4.0, VBMaxLCD.dll provides all the functionality of a third-party control but without the overhead.

Not only that but you get the source code too, so you can see how it works and can tweak it any way you want to suit your own needs.

Registration

This is a shareware utility which you can register for US \$5 per copy. Please send payment to:

Mike Stanley
25 Lansdowne Street
Manchester
NH 03103
USA

In return, I will give you a registration ID and an encryption key which will allow you to unlock the source code from the file 'VBMaxLCD Source Code.grk'—**don't forget to include your Internet e-mail address**. Better still, also send me an e-mail to let me know that your registration is on the way.

Any future updates to the software will use the same encryption key, so you can always get them at no additional cost.

If you don't have an e-mail address, you can receive the registration ID and encryption key via snail-mail by sending

me a postage-paid self-addressed envelope.

Important—Use at Your Own Risk!

While every effort has been made to ensure a reliable, high-quality product, you should note that this software is provided 'as is' without any kind of warranty whatsoever, neither explicit nor implied.

In order to use this software, you must agree 100% that I will not be held liable in any way, shape or form (not even a little bit) for anything untoward that befalls anyone or anything, either directly or indirectly, as a result of using this software no matter how calamitous, disastrous or inauspicious the occasion may be.

Your use of this software constitutes acceptance of these terms.

Installation

Copy the files to a folder of your choice and edit Reg.bat so that it contains the correct path for the regsvr32.exe program that comes with VB 4.0. Please note that regsvr32.exe is not copied to your hard-drive automatically when you install VB – you can find it in the Tools\Pss folder on your VB CD.

Run the edited Reg.bat which will update the Windows registry so that it knows about VBMaxLCD.dll.

Go into VB, load Demo.vbp and run it.

If you want to use the DLL in your own programs, you have to add a reference to it as follows:

From VB, select Tools from the menu.

Then select References...

Locate VBMax Liquid Crystal Display in the list and click the check box so that an X appears in it. Click OK.

The OLE server is now visible in the Object Browser.

Frequently Asked Questions

Q. I try to run your demo program but it keeps telling me that OLE Automation can't create the object. Why?

A. You have not registered VBMaxLCD.dll with Windows. Please refer to the installation instructions above.

Q. I have searched my hard-drive but cannot find regsvr32.exe – what's the scoop?

A. Regsvr32.exe is not copied to your hard-drive automatically when you install VB – you can find it in the Tools\Pss folder on your VB CD.

Q. Wouldn't it be easier if you just created a regular installation program?

A. One thing I don't like about VB 4 is the amount of extra baggage that must be included with the installation of even the simplest programs—this translates to long download times from online services. Seeing as VBMaxLCD.dll is meant for programmers who already have all the necessary software installed on their computers, I decided that manually installing the DLL was a fair tradeoff to racking up your online connect time.

Technical Support

Technical support is available via e-mail. Please send questions, comments, criticisms, etc. to 74632.2227@compuserve.com.

Gronked Files

These are encrypted files, created using the VBMax Gronk Meister, that contain one or more other files. In this case, the file 'VBMaxLCD Source Code.grk' contains the complete, commented VB 4.0 source code for VBMaxLCD.dll.

The source files may be extracted using the included VBMax Degronker utility. Before you can do that, however, you need to know the encryption key. This key will be given to you when you register VBMaxLCD.dll.

Packing List

Read Me.rtf	-	The document you are reading.
Demo.vbp (and related files)	-	Source code for demo program showing the DLL in action.
Reg.bat	-	Registers the DLL with Windows (you have to edit the path).
Unreg.bat	-	Unregisters the DLL with Windows (you have to edit the path).
VBMaxLCD Source Code.grk	-	Complete, commented source code for the DLL in gronked format.
Degronker.exe	-	Utility to extract the source code from the gronked file.

How to use the DLL

When you include a reference to VBMaxLCD.dll in VB as described earlier, you will have access to a class called clsLCD. When you select this class using the Object Browser, you will see the methods and properties available to you. You can see a brief description of what each one does by clicking on them.

In your application, you can create one or more objects from this class module. You need a separate object for each LCD control you want to create.

Objects are created from this class by using code like this:

```
Dim moLCD As New clsLCD
```

In the following methods and properties summaries, *Object* evaluates to an object created from clsLCD. In this example it would be moLCD.

You create an LCD control by drawing a PictureBox control on a form and then passing it to the Container method in the object created by clsLCD.

The DLL provides a high level of control over the appearance of the LCD and also gives you the means to blink colons, flash digits and hide or show the unlit segments of the digits. The demo program includes many examples.

Methods Summary

Object.About

Displays an About box.

Example: moLCD.About

Object.Cls

Clears the contents of the LCD.

Example: moLCD.Cls

Object.Redisplay

Redisplays the current caption. If you need to change the size of the PictureBox container, use this method afterwards to redisplay the LCD at the correct size.

Example: `moLCD.Redisplay`

Object.SelectBackColor

Opens up a common dialog color selection window and allows you to select a color for the background.

Example: `moLCD.SelectBackColor`

See also the BackColor property and the notes on color selection.

Object.SelectForeColor

Opens up a common dialog color selection window and allows you to select a color for the digits.

Example: `moLCD.SelectForeColor`

See also the ForeColor property and the notes on color selection.

Properties Summary

Object.Alignment = Setting

Sets or returns a value that determines how the digits are aligned in the PictureBox container.

Setting One of the following:

- 1) `gnALIGN_LEFT`
Left-justifies the digits in the PictureBox.
- 2) `gnALIGN_RIGHT` (default)
Right-justifies the digits in the PictureBox.
- 3) `gnCENTER`
Centers the digits in the PictureBox.

Example: `moLCD.Alignment = gnALIGN_RIGHT`

Object.Autosize = Setting

Sets the Autosize behavior.

Setting One of the following:

- 1) `gnAUTOSIZE_CONTAINER_TO_LCD` (default)
Adjusts the size of the PictureBox container to match the size of the digits.
- 2) `gnAUTOSIZE_LCD_TO_CONTAINER`
Adjusts the size of the digits to match the size of the container.

3) gnAUTOSIZE_OFF

Makes no adjustment to either the container or the digits.

Example: `moLCD.AutoSize = gnAUTOSIZE_CONTAINER_TO_LCD`

Object.BackColor [= Color]

Sets or returns the background color of the LCD.

Color: A value or constant that determines the background color of an LCD.
The default color is black.

Example 1: `moLCD.BackColor = vbBlue`
Example 2: `moLCD.BackColor = &H8000&`
Example 3: `IColor = moLCD.BackColor`

See also the `SelectBackColor` method and the notes on color selection.

Object.BlinkColon [= Boolean]

Sets or returns a value that determines whether colons contained in the display should blink.

Boolean: An expression that evaluates to True or False.
The default is False.

Example 1: `moLCD.BlinkColon = True`
Example 2: `bBlink = moLCD.BlinkColon`

This property is provided so that clock type controls can have blinking colons to mark the seconds. If this property is True, the LCD object will alternately show and hide any colons in the display. It is the responsibility of the application to set a timer to update the caption property at the appropriate intervals. An interval of 500 milliseconds will cause the colon to blink about once a second (half a second visible and half a second invisible).

Object.Caption [= String]

Sets or returns a value that determines the contents of the LCD.

String A string expression that evaluates to the digits displayed as the caption.

Example 1: `moLCD.Caption = "123,456.78"`
Example 2: `moLCD.Caption = Format$(Now, "mm-dd-yy")`
Example 3: `moLCD.Caption = moLCD.Caption + 1` (Evil Type Coercion)

Set Object.Container = PictureBox

Mandatory property that specifies the PictureBox used for the LCD.

PictureBox Regular VB PictureBox control.

Example: `Set moLCD.Container = picLCD`

Object.FlashDigits [= Boolean]

Sets or returns a value that determines whether the entire display should flash on and off.

Boolean: An expression that evaluates to True or False.
The default is False.

Example 1: moLCD.FlashDigits = True
Example 2: bFlash = moLCD.FlashDigits

This property is provided so that controls can flash to draw attention to themselves when an event occurs. Examples could be an alarm clock that flashes when the designated time is reached or a nuclear power plant control that flashes when a core meltdown is imminent.

Object.ForeColor [= Color]

Sets or returns the color of the digits in the LCD.

Color: A value or constant that determines the digits color of an LCD.
The default color id red.

Example 1: moLCD.ForeColor = vbBlue
Example 2: moLCD.ForeColor = &H8000&
Example 3: lColor = moLCD.ForeColor

See also the SelectForeColor method and the notes on color selection.

Object.ShowUnlitSegments [= Boolean]

Returns or sets a value that determines whether the unlit segments of the digits are displayed.

Boolean: An expression that evaluates to True or False.
The default is False.

Example 1: moLCD.ShowUnlitSegments = True
Example 2: bShowUnlit = moLCD.ShowUnlitSegments
Example 3: moLCD.ShowUnlitSegments = Not moLCD.ShowUnlitSegments

Showing the unlit segments works better with some color combinations than others. For instance, red digits against a black background look pretty good but change the digits color to white and the effect is not so good. You will have to experiment to see which combinations work for you. Also, you really need 256 colors or higher for a decent effect.

Notes on Color Selection

There are no built-in restrictions about what colors you can use for the digits or the background. Realistically, however, you have to exercise a little caution because of potential differences between the color palette settings of your system and those of the computers on which your application may run.

If your system is running in 16 bit or higher color mode, the colors you choose may look a lot different on systems running in 256 colors. You could end up with dithered colors which really don't look that great with LCDs. A safe approach may be to select solid colors only while you have your system running in 256 colors.

Mike Stanley
New Hampshire

USA

E-Mail: 74632.2227@compuserve.com