ObjectSet Custom Controls

ObjectSet Custom Controls for Visual Basic

Version 1.0

October 1994.

(C) Copyright by Q-LOGIC

PO Box 207
Casula Mall
Casula
NSW 2170
Australia
Phone: (61) (02) 821-4121
Fax: (61) (02) 821-4121
Compuserve: 100355 , 207
msn: GO QLogic

Thank you for trying Q-LOGICs ObjectSet custom controls. Please feel free to send us any comments you have, or ideas for improvements. All feedback is much appreciated!

Copyright Notice

This software is copyrighted software. The copyright is owned exclusively by Q-LOGIC. To use or distribute this software for purposes other than evaluation you should obtain a license to do so from Q-LOGIC. A license may be purchased from Q-LOGIC for only **$10** (Australian)**.**

Shareware Distribution

To enable this software to be properly evaluated prior to purchase, this software is distributed freely on a Try-Before-You-Buy basis, commonly known as SHAREWARE. *(Note that SHAREWARE is a method of distribution, and not a type of software. Q-LOGIC endeavours to produce its software to the highest commercial standards, regardless of what method is employed to market it.).* You may copy and distribute this software free of charge provided it is for the purpose of evaluation only.

Please Help Support Shareware

Shareware enables high quality software to be delivered to customers at the lowest possible prices, devoid of marketing expenses and other costly overheads. Please help support this concept by paying for software you wish to keep. If those who spend much time and effort to create this software are not paid, the outcome is obvious - they will stop doing it.

ObjectSet Custom Controls

Q-LOGIC
PO Box 207
Casula Mall
Casula
NSW 2170
Australia
Phone: (61) (02) 821-4121
Fax: (61) (02) 821-4121
Compuserve: 100355 , 207
msn: GO QLogic

Purchasing A License

The standard license costs only **$10** (Australian**)** per person. This license enables the license owner to use the **ObjectSet** custom controls to develop applications, and then to distribute the said controls *(not the anciliary programs or this help file)* with those applications **royalty free.**

Businesses and Institutions wishing to obtain licenses should contact Q-LOGIC direct to negotiate a site license, or else purchase one of more ordinary licenses for each individual within the organisation who is to use the **ObjectSet** software.

To obtain a license to use the **ObjectSet** custom controls), call, fax or send email to Q-LOGIC (see above), and Q-LOGIC will mail out the license agreement.

You can order your license via phone, fax, e-mail or by post. When ordering via the phone, fax or e-mail payment must be made by American Express. If ordering by letter, payment may be made by American Express or by check or money order. If paying by American Express, make sure to quote the expiry date as well as the card membership number.

License Only

Note that we do not send a copy of the software itself, *only the license agreement.* The software is distributed via the shareware mechanism, which is the full and uncut version of the software. Sending the software on a disk would inevitably add to the cost **(which I am trying to keep down)**, especially for overseas purchasers. (*It is assumed you have already obtained a complete copy of the software to be able to view this document*.) If for any reason you are unsure as to the quality, completeness or age of the **ObjectSet** software that you currently have, first try and obtain another copy of the software from Compuserve, a local bulletin board or a friend. If you are unable to obtain a copy that you are happy with, then contact Q-LOGIC direct at the above address.

ObjectSet Custom Controls

Contents Of Object Set Custom Control Package

This windows help file (QOBJSET.HLP) is part of the Q-LOGIC Object Set custom control package. The package consists of the following:

1. QOBJSET.HLP (*This file*).
2. QMET.VBX.
3. QINT.VBX.
4. QSTR.VBX.
5. QFLT.VBX.
6. QOBJSET.MAK.
7. FORM1.FRM.
8. FORM2.FRM.

If for any reason you have not obtained all of these files, please contact the retailer who supplied the **ObjectSet** software, or contact Q-LOGIC at the above address.

The custom controls themselves comprise the four above VBX files, namely QMET.VBX, QSTR.VBX, QINT.VBX, and QFLT.VBX. All four controls are completely independent, and require no additional license files or DLL files to function. All the other files provided with this package are there to provide information on how to correctly use the controls. ***(Note that only the VBX files can be supplied with an application (****see the later section on Copyright for more information)****).***

QLogic

ObjectSet Custom Controls

Installing The Software

Simply copy the four VBX files to the Windows System directory (usually C:\WINDOWS\SYSTEM). They are now ready to include in any Visual Basic project. Do not forget you will need to add them to the project using the Add File option on the File menu.

You need only retain the controls you wish to use. The four controls are completely independant, and it may be quite likely that you will not require the floating point control. In that case, to save space simply delete it.

Running The Demo Program

The demo program QOBJSET.MAK can be left in whatever directory you have unzipped the controls. Simply load Visual Basic and select Open Project from the File menu, and select QOBJSET.MAK. Select Run to run it. The demo program is intended as a introduction to using the controls and its main benefit lies in examining the code it contains, rather than in running it.

To make the demonstration program attractive, two 256 color bitmap images have been included. As these use up a significant amount of disk storage (approx 200 Kbytes) *it is recommended that you delete the demo program* as soon as you become familiar with the controls. If you wish to keep it you can reduce the amount of disk storage required by deleting the images or by replacing them with 16 color images, which use half the space. The images are not essential to the correct functioning of the demo program.

Removing The Demo Program

To delete the demo program, simply delete the following files:- QOBJSET.MAK, FORM1.FRM, FORM2.FRM.

ObjectSet Custom Controls

The Purpose of OBJECTSET Custom Controls

The Visual Basic language is not truly object oriented and does not support Classes and Methods as does C++. Its true that custom controls have some of the characteristics of true objects (although these lack method type function calls, with virtually all actions being triggered by setting properties), but they are of course written in C and not in Visual Basic.

Visual Basic does come annoyingly close to providing true objects with its forms, which share many of the most important charactersitics of objects. They can be instantiated dynamically (one or more instances of a form can be created at run time) in the same way that true objects can. They do provide encapsulation, (that is to say the internal workings of the form are hidden from external code) which any true object must, but they do so to the extreme!. It is impossible to access the variables or functions of a form from code outside the form.

It is possible to access the custom controls on a form from outside the form, and so you can provide indirect access to the variables on a form using text boxes, and you can trigger functions on the form by using the change event of a text box. These techniques do work, but they are not very efficient, and the coding tends to be cumbersome. Imagine some 10 or more text boxes on a form, each one holding a single integer value, trying to pretend they are integer variables? Very few people would attempt such coding and so the potential of the form as an object is wasted.

The **ObjectSet** custom controls come to the rescue. They have been designed specifically to provide pseudo form level variables (QINT, QFLT and QSTR) and a method interface for calling functions (QMET).

1. They are graphical controls and so use no Windows GDI resources.
2. They are small and have the minimum number of properties, and so use little memory.
3. Each control has a single main property, which is its default property. Thus the **property name does not need to be coded** when setting the property, resulting in clean code.
4. Few properties means less setup time.

The **ObjectSet** custom controll set consists of four controls, QMET, QFLT, QSTR and QINT. These are all low-resource Graphical type controls designed to provide the ability to call private form functions from another form and to pass data to and from the form as if it had visible variables.

Programming Using Form Objects

Coding programs using the **ObjectSet** custom controls involves thinking of and using VB forms as objects, and the functions and subroutines on that form as methods. All data and code relating to this object are encapsulated within the form. You do not code any global variables and no BAS module is used. All variables are kept within the form and are private to the form. No subroutines or functions are declared in BAS modules either, but are also kept within the form.

The objects methods become subroutines or functions within the form, and these are invoked using the QMET. Method control. Any required parameters for these functions (methods) are passed using QSTR(string), QFLT(integer) or QFLT(floating point) controls placed on the form. By communicating with the form via the **ObjectSet** controls (and possibly also list box and pocture box controls) the form becomes a single coherent entity which can be reused, and therefor included in any future projects, without having to duplicate code. Also, since the form has no global elements, *there are no data name conflicts* to pose problems when many disparate modules are brought together within a single project.

One of the greatest advantages of using form objects is that a form can be loaded mutliple times at run time by using form variables and the NEW keyword. This is very difficult to achieve using global variables without proliferating arrays all over the place. This only exacerbates the problem of data name conflicts, especially within large programs.

ObjectSet Custom Controls

QMET **QINT** **QSTR** **QFLT** **Properties** **Events**

QMET Method Control

This is the method control which is used to trigger functions within a form. It has a default property METHOD which need not be coded. To trigger a function, simply set the control to a string representing the function to call, as follows. The example assumes a QMET control has been placed on a form called Form2 and the control itself has been called ACTION.

```
FORM2!ACTION = "get customer"
```

The QMET control has a single event called METHOD, which is invoked whenever the default property METHOD is set. The METHOD property string is passed to the event function as a parameter mName. (You could refer to the property directly, but it is quicker to access a variable which is why the parameter is there). Simply code a case statement in this event subroutine with an entry for each function you wish to call, as follows:

```
sub Action_Method (mName as string)

select case mName
case "get customer"
        do_get_cust
        exit sub
case "add customer"
        do_add_customer
        exit sub
end select

end sub
```

In addition to setting the default property, you can trigger the METHOD event by executing a REFRESH method against the control; e.g.

```
form2!Action.refresh
```

QINT Integer Control

This is the Integer variable control which is used to pass long or short integer values to a form, and acts like a form level long integer variable. Place as many on the form as you need. The default property which holds the integer value is called INTEGERVALUE, but it never needs to be coded. The following example shows how we pass values to a form called Form2 which has three QINT controls called Year, Month and Day.

```
form2!Year = 1994
form2!Month = 7
form2!Day = 28
```

The QINT control supports a single event called REFRESH , triggered when a REFRESH method is executed against the control; i.e.

```
form2!Year.refresh
```

QFLT Floating Point Control

This is the single variable control which is used to pass single sized floating point values to a form, and acts like a form level single variable. Place as many on the form as you need. The default property which holds the single value is called FLOATVALUE, but it never needs to be coded. The following example shows how we pass values to a form called Form2 which has two QFLT controls called MouseX and MouseY

```
form2!MouseX = X
form2!MouseY = Y
```

The QFLT control supports a single event called REFRESH , triggered when a REFRESH method is executed against the control; i.e.

```
form2!MouseX.refresh
```

QSTR String Control

This is the string variable control which is used to pass strings to a form, and acts like a form level string variable. Place as many on the form as you need. The default property which holds the string is called STRINGVALUE, but it never needs to be coded. The example shows how we pass strings to a form called Form2 which has three QSTR controls called Name, Address and Country.

```
form2!!Name = "John Williams"
form2!Address = "28 George Street, Sydney"
form2!Country = "Australia"
```

The QSTR control supports a single event called REFRESH , triggered when a REFRESH method is executed against the control; i.e.

```
form2!Year.refresh
```

Note that the QMET control also passes a string and can be used in place of a QSTR control. However, the QMET control's Method event is always triggered each time a new string is placed into it. This will have some impact on speed. Otherwise, the results will be largely the same.

Property Reference

QMET   Left   Top   Method

QINT	Left	Top	IntegerValue
QSTR	Left	Top	StringValue
QFLT	Left	Top	FloatValue

Event Reference

QMET	Method
QINT	Refresh
QSTR	Refresh
QFLT	Refresh

The Left and Top properties are the standard VB properties and allow the controls to be positioned at design time. Since the controls are always invisible at run time, these properties are never used in code.

ObjectSet Custom Controls

Copyright

The **ObjectSet** custom controls (QMET.VBX, QSTR.VBX, QINT.VBX, QFLT.VBX), the QOBJSET Visual Basic demo program (QOBJSET.MAK and its associated files), and this document (QOBJSET.HLP) are all copyright of Q-LOGIC. They may be used without a license **only for the purpose of evaluation.**

When licensed, the custom controls **only** (QMET.VBX, QSTR.VBX, QINT.VBX, QFLT.VBX) may be distributed ***royalty free*** as part of an application. The recipient of said application containing the **ObjectSet** custom controls is not licensed to use them in design mode, nor to further distribute them as part of another application separate from the application from which they were obtained. Anyone wishing to use the Object Set controls in design mode **must** obtain a license.

The other support files provided with the Object Set package (QOBJSET.HLP - this text file, QOBJSET.MAK, FORM1.FRM, FORM2.FRM) **must not be distributed** and are for use only by the licensed owner.

ObjectSet is the name given to the four custom controls designed to facilitate communications between any Visual Basic code and a Form object. The controls are as follows:-

QMET.VBX. Method control, QINT.VBX.   Integer control, QSTR.VBX. String control, QFLT.VBX. Floating point control.

Only the QMET control supports this event. The Method event is triggered by setting the Method property to any value. The Method property is then passed to the Method event as a string parameter. The Method event can also be triggered by using the Refresh method against the control. In that case, the previous Method property is passed to the Method event.

All controls **except** the QMET control support the Refresh event. The event is triggered by executing the Refresh method against the control. No parameters are passed to the Refresh event.

The QMET control supports this property. Setting this string property to any value will trigger the Method event. The Method property is passed as a parameter to this event.

The three controls QINT, QFLT and QSTR each have a single value property which are IntegerValue, FLoatValue and StringValue respectively. Because these properties represent the default property for each control, you need never use these names in code to reference the property. This results in cleaner looking code.

QOBJSET.HLP is the Windows Help file you are currently viewing. This file is part of the ObjectSet package and should always be copied together with all the other components.

The only credit card that Q-LOGIC accepts for TeleMarketing purposes at present.

Design mode - where the custom controls are used within a program development environment such as Visual Basic, Visual C++, or any programming system which can play host to VBX controls, for the purpose of program development. Only the licensed ObjectSet user is permitted to do this.