Quirk

Quirk Engine Version 1.10
© Copyright 1999 Inscitz Software
FULL Release (Freeware)
8th May 1999

http://members.xoom.com/inscitz

Developers Documentation

Table Of Contents

1.1 : Terminology

'Product' in this document is defined by the Quirk engine, its related files (including this documentation) excluding the software the Quirk engine is incorporated in.

'Users' in this document is defined by the user(s) of your (the developer) software that incorporates the Quirk engine.

1.2 : Copyright / Legal Information

You, the developer, and the users of your software comply that use of this product signifies that you fully understand and agree to the following terms and conditions listed below;

•No responsibility is taken for any damage / problems that this product has/may incur. You fully understand that you use the engine at your own risk.

•All trademarks and registered trademarks used in this product are property of their respective owner(s).

•This product remains the copyrighted property of Inscitz Software. You may not modify this product in any way (decompile, edit, remove parts).

•You may not sell this product by itself or in a manner that does not incorporate the Quirk engine into software without the prior consent of Inscitz Software. Developers that incorporate the Quirk engine must read section ' 1.5 ' (Royalty / Distribution) for the distribition agreement.

•Users of your software are made fully aware of this section before the Quirk engine is used. For example, in your software's installation routine, include a 'legal section' that appends this section which is unmodified / readable and directed only at the use of this product.

•No warranty and 'Technical Support' of any kind is provided with this product.

Failure of agreement or understanding of the above terms and conditions shows that you are not permitted legally to use this product.

-- End of Copyright / Legal section --

1.3 : Introduction
===================================

First of all, thank you for choosing to use Quirk.

Secondly, this is a non profit based product, so theres no fee obligations or cash royalties.

Thirdly, here is what the products all about...

Quirk, is a engine that incorporates itself into a Visual Basic program (project). With the Quirk engine, your program can access certain functions and settings not found with Visual Basic itself. The Quirk engine is an OCX (Custom Control).

Quirks functions are described in the provided sections in this document but a basic overview is that it lets you add really cool goodies such as;

Because Quirk is changing all the time, more functions are becoming available. If you need a specific function, and Quirk has not got it, e-mail Inscitz software at

insctz@hotmail.com. Functions range from system & file, sound, internet, display, variable and much more more.

1.4 : Requirements

==

This product requires the following to operate in a functional status:

- **Windows 95/98 (NT is not recommended but should work)**
- **8MB Ram**
- **This documentation**
- **A PC of course, and if it runs Windows, its good enough**

1.5 : Royalty / Distribution Agreement

===

To distribute the Quirk engine, you must meet the following agreement:

- **You and the users of your software must read, fully understand and agree to the terms and conditions of use of quirk (Section 1.2).**
- **You do not sell or lease the Quirk engine by itself.**
- **The Quirk.Doc file is distributed with the Quirk.OCX file.**
- **The Quirk.OCX file is placed under the user's window\system folder.**
- **You are distributing the latest version of Quirk (available from our website)**
- **You e-mail us at inscitz@microed.com.au and tell us the name of your software you are distributing with Quirk, the version of your software, where to download the software, your software's (or company's) URL (Internet Address), and finally your contact e-mail address.**

If you comply with the above terms and conditions, you may freely distribute the Quirk engine with your software.

There is no royalty in charge for using Quirk.OCX. You may sell the quirk engine as long as it is distributed as an addin to your software.

1.6 : Version History

==

Here is a list of what bugs have been splat, what words have been missspelled and what ever else we did to update it;

1.00 5th May 1999

First release

1.01 7th May 1999
New Function: GetComputerName
New Function: GetWindowsDir
New Function: GetSystemDir
New Function: GetTempDir
New Function: GetWindowsVer
Fixed: Bug which prevented upgrading of control

1.10 12th May 1999
Quirk was completly rewritten
Any app using version 1.0 or 1.01 will have to rewrite their code
Sorry about this, we just had to rewrite the Quirk engine
It wont happen again.
New Function: GetComputerUserName
New Function: GetKeyboardType
New Function: GetRAMTotal
New Function: GetRAMUsed
New Function: GetRAMFree
New Function: GetDriveFileSystem
New Function: GetDriveSerial
New Function: GetDriveVolume
New Function: SetDriveVolume

be sure to keep checkin the website for new versions with even more functions.

1.7 : Where to find the latest version of Quirk
===

Here is a list of internet sites where you can download the latest version of Quirk;

INSCITZ SOFTWARE HOMEPAGE:
www.members.xoom.com/inscitz

At the top of this file is the release date, if it was more than a month ago, there will most likely be a newer version of Quirk available.

There are currently two packages of Quirk:

1. Quirk Engine Package
The Quirk.Doc and Quirk.Ocx file that install/uninstall (Recommended for upgrading only)

2. Quirk Developers Package

Same as #1, plus it includes demonstration projects to help you get using Quirk Engine. You may freely export the code from the demonstration projects into your own.

1.8 : Reporting problems with Quirk

==

Even Windows occasionally kicks the bucket, so its also possible for Quirk to find a bug that we didn't. If you happen to come across a Runtime error or a dialog that has a 'Oh no! A Bug!' caption or something does not function the way its supposed to, PLEASE, tell us. Please send an e-mail to us at:

inscitz@microed.com.au

1.9 : Quirk Package / File List

==

Here is a list of what is distributed when you download the developers package from one of site(s) in section 1.7;

•Quirk.ocx	=> The Quirk engine itself
•Quirk.doc	=> This documentation
•Demo.zip	=> Quirk Demostration project

The Quirk demonstration project is a Visual Basic 5 (or better) project that illustrates all of Quirks features, with lots of comments on how to use the functions of Quirk to their full potential. You can copy/paste code out of the Quirk Demonstration project into your own if its of help.

2.0 : Note to the developer

====================================

Greetings!

Thanks for using Quirk. Quirk will not have fee's attached to it so dont ever worry about having to pay for use of Quirk in your software. It does not downgrade your software excepting for our loading dialog which appears when you initialize the control which gathers system information and details on the local PC.

I hope that you find Quirk of great use for your software and would love to hear comments / suggestions / idea's on new functions that may be of use to you / bug reports / fan mail (ha ha, only joking).

2.1 : Setting Up Quirk

=====================================

In your Visual Basic project, simply hold Ctrl + T to bring up the components list.

Find the 'Quirk Engine' in the list and ensure its ticked (if its not, click it!)
If Quirk is not in the list, browse for it with the browse button (you should know where you put the thing but if you cant find it, it should be in your windows\system folder). Once you locate it, click Ok and it will add it to the list! Now make sure it has a tick beside 'Quirk Engine'.

Click Ok in the Components list to add the control to your toolbar (the thing where the picture box, label, textbox, timer controls are found). Its symbol is a folder with some tools on it, and its tooltip says 'Quirk'.

Place the Quirk control on your form my selecting it from the toolbar and drawing it on the form or if your too lazy, just double click on it.

NOW... heres the tricky bit

Quirk does not initialize straight away. This is because your program may want to do certain things first without the delay of loading the control. You will need to initialize the Quirk control before you can use its commands. You initialize it by placing this simply line of code anywhere you want in your project, as long as its executed before any other commands are sent to Quirk (preferrably in the loading sequence);

```
Quirk1.Initialize
```

Quirk1 is the default name for the control, you may however of renamed the control after placing it onto your form, if so, use the name you gave it.

The Quirk control does not need to be Unitialized or anything, and if you happen to initialize it again, it will ignore initialization as it has already taken place.

2.2 : Functions / Commands

=====================================

Here is a full list of what the functions / commands in Quirk do (listed in categories);

Total Functions: **27** Shown in order

Function	Section to find it in (and it ifs got no section, its the section above it)
Initialize	Quirk Related
DisplayGetRes	Display Related
DisplayGetResString	
DispaySetRes	

FixCase	Variable Related
FixValue	
FileExist	File / Folder Related
DosDeltree	
GetFileSystem	
GetDriveSerial	
GetDriveVolume	
SetDriveVolume	
DisableTaskBar	Security Related
DisableTaskSwitching	
ForceLogOff	
ForceReboot	
ForceShutDown	
GetComputerName	System Related
GetComputerUserName	
GetWindowsDir	
GetSystemDir	
GetTempDir	
GetWindowsVer	
GetKeyboardType	
ShowMsg	Miscellanious Related

QUIRK RELATED

.Initialize

This function initialize the Quirk engine.

Syntax:

Initialize ( )

This command is required to operate any other function in Quirk. The initialize function needs to be called only once but if it is called more than once, it will be ignored to prevent any code slowdown.

Tips:

Ensure this function is used before any other Quirk function otherwise the other functions will return an error.
Anotherwords ensure you have it in your 'Form_Load' section or 'Sub_Main'.

DISPLAY RELATED

.DisplayGetRes

Returns an integer value representing the current display resolution.

Syntax:

DisplayGetRes ( )

Here is a list of the integer values that could be returned:

0	Other
1	320 x 200
2	640 x 400
3	640 x 480
4	800 x 600
5	1024 x 768

Tips:

Use the DisplayGetResString to return a string ("320 x 200" etc).

.DisplayGetResString

Returns a string value of the current display resolution.

Syntax:

DisplayGetResString ( )

Here is the style of string is produced by the return of this:

640 x 480

.DisplaySetRes

Sets the display resolution to the given value mode

Syntax:

DisplaySetRes (*ResMode as integer*)

Resmode can be a value ranging between 1 and 5;

1	=	320 x 200
2	=	640 x 400
3	=	640 x 480
4	=	800 x 600
5	=	1024 x 768

Tips:

If ResMode does not range between 1 and 5, an error (1001) is shown.

VARIABLE RELATED

.FixCase

This function returns a string which each letter of each word is capitalized and the others are lower case.

Syntax:

FixCase (*stringtochange as string*)

The stringtochange string will be changed to be upper case for each starting letter of each word and lowercase for each other letter. eg. "JOHN BARRY!" would return 'John Barry!"

Tips:

This function is useful in the loosefocus event of textboxes, which will fix the style of text before it is saved to file or a variable. eg. *Text1.Text = Quirk1.FixCase(Text1.Text)*

.FixValue

This function returns a variant value which is a number with comma's and fixs the number of values after the decimal point.

Syntax:

FixValue (*CurrencyValue as variant, RightOfDecimalPos as integer, [UseCommas as boolean]*)

The CurrencyValue can be any number value that will be returned as the new variant. The RightOfDecimalPos can range from 0 to 2 which will determine how many decimal values are given to the right of the decimal point. The UseCommas is optional, if not set, it defaults to false. UseCommas determines wheather commas will be placed at the thousands, millions (eg. 1,234,567.00).

Tips:

This function is good for displaying or setting an amount of money. For example, if the user is prompted to enter a cash number

(1042), you could save this as "$1,042.00" by using the following technique:
(Let CashValue be set to 1042)

NewStringValue = "$" & Quirk1.FixValue(CashValue, 2, True)

or to display a message with the fixed value;

Quirk1.ShowMsg ("$" & Quirk1.FixValue(CashValue, 2, True))

FILE / FOLDER RELATED

.FileExist

This function returns a boolean value of the existance of a file.

Syntax:

FileExist (*filename as string*)

The filename is the location such as "c:\io.sys" of the file to check for existance. If the filename string is empty (""), the function will return false (not found).

Tips:

Before opening a file, its always worthwild to ensure it exists rather than receiving a runtime error.

.DosDeltree

This function deletes all folders and files in and under a specified foler. Returns true or false if successful.

Syntax:

DosDeltree (*Path as string*)

The Path is the location of the folders and files you wish to delete. If no path is supplied, returns false.

Tips:

I think this function has problems with it, so try it out if you would.

SECURITY RELATED

.DisableTaskBar

This function enables / disables the Windows Taskbar

Syntax:

DisableTaskBar (*TrueOrFalse as boolean*)

Set TrueOrFalse to true to enable the taskbar, or false to disable the taskbar. The system tray is also disabled.

.DisableTaskSwitching

This function enables / disables Windows Taskswitching (Ctrl+Alt+Del) and (Alt+Tab).

Syntax:

DisableTaskSwitching (*TrueOrFalse as boolean*)

Set TrueOrFalse to true to enable Taskswitching, or false to disable Taskswitching.

Tips:

Its always a good idea to ensure its re-enabled on closing of your app, so a smart idea would be to place a line of code in your Form_Unload( ) section to enable Taskswitching.

.ForceLogOff

This function logs off the current logged on user and brings up the Windows log on dialog.

Syntax:

ForceLogOff ( )

Dont forget, this closes all open applications, the explorer and brings up the log on dialog.

Tips:

If you dont have a log on scheme on your PC, then Windows will just restart the Explorer. All open applications stay active.

.ForceReboot

This function logs off the current logged on user and closes all open applications, exits windows and restarts the computer.

Syntax:

ForceReboot ( )

Dont forget, this closes all open applications, the explorer and any work not saved will be lost, so ensure its safe first!

Tips:

This bypasses the sound scheme that is played on exiting windows however, which speeds things a long a little.

.ForceShutDown

This function logs off the current logged on user and closes all open applications, exits windows and shuts down the computer.

Syntax:

ForceShutDown ( )

Dont forget, this closes all open applications, the explorer and any work not saved will be lost, so ensure its safe first!

Tips:

This bypasses the sound scheme that is played on exiting windows however, which speeds things a long a little.

SYSTEM RELATED

.GetComputerName

Returns the name of the Computer (Control Panel/Networks)

Syntax:

GetComputerName ( )

Returns an empty string if an error occurs or no name is set to the computer.

.GetComputerUserName

Returns the name of the user on the Windows OS

Syntax:

GetComputerName ( )

Returns an empty string if an error occurs or no name is set to user.

.GetWindowsDir

Returns the name of the folder that is the Windows Directory.

Syntax:

GetWindowsDir (*[InsertBackSlash as Boolean]*)

Set InsertBackSlash (Not Required: Default is false) to true if you want a backslash ("\") included in the returned string.
Returns an empty string if an error occurs.

.GetSystemDir

Returns the name of the folder that is the System Directory.

Syntax:

GetSystemDir (*[InsertBackSlash as Boolean]*)

Set InsertBackSlash (Not Required: Default is false) to true if you want a backslash ("\") included in the returned string.
Returns an empty string if an error occurs.

.GetTempDir

Returns the name of the folder that is the Temporary Directory.

Syntax:

GetTempDir (*[InsertBackSlash as Boolean]*)

Set InsertBackSlash (Not Required: Default is false) to true if you want a backslash ("\") included in the returned string.
Returns an empty string if an error occurs. This is not the DOS temporary folder (in the DOS environment), but the temporary folder for Windows used by the internet, and software.

.GetVersion

Returns an integer representing the Windows Version

Syntax:

GetVersion ( )

Returns an integer of either:

3	=	Windows NT 3.51
311	=	Windows 3.11
4	=	Windows NT 4
95	=	Windows 95
98	=	Windows 98
-1	=	Unknown

.GetKeyboardType

Returns the name of the Keyboard

Syntax:

GetComputerName ( )

Returns a string with 'Unknown' if the keyboard is unknown.

MISCELLANIOUS RELATED

.ShowMsg

This function uses the MSGBOX function, but uses a default style scheme of an exclamation icon, and an Ok button.

Syntax:

ShowMsg (*Message as string, [Title as string]*)

The Message is exactly that. It is what text will be displayed in the message dialog. The title is optional, it not set, it defaults to "Message".

Tips:

Use the ShowMsg function to display a quality message dialog quickly.

2.3 : Future Functions
====================================

This section contains a list of functions that are we are thinking about implementing in the near future. We are also looking for idea's from you about what type of functions could be added or which ones could be improved to make Quirk better;

- DosXCopy
- Obtain / Set Display Colour Resolution
- Obtain / Set Main audio Volume
- Obtain / Set Wave audio Volume
- Obtain / Set CD audio Volume
- Obtain Free Disk Space / Total Disk space
- Determine Drive type (HDD, FDD, CD-ROM, NETWORK)
- Advanced ShowMsg box, that has animated pics, the choice of setting a default title, change the keys shown, and its mode.
- Constants for the display mode - more reliable in the future as well
- Clear Documents folder (Start Menu)
- Clear Favourite folder (Start Menu)
- Send file to Recycle bin
- Copy Files (Brings up the copying files dialog)
- Activate Window (hwnd or title) - have two seperate commands

2.4 : Quirk Errors
====================================

This section contains a list of errors that you might receive relating the improper use of Quirk.

Error Code: 1000
Error Description: Quirk function called when Quirk has not been initialized!
Error Resolution: Call the '.Initialize' function to set up Quirk before calling any of Quirk's functions.
Error Example Fix:

```
Private Sub Form_Load( )
Quirk1.Initialize
' The Quirk initialize line needs to be used before any other Quirk
' function in your code.
Quirk1.ShowMsg("Error Code '1000' won't occur anymore")
```

Error Code: 1001

Error Description: Value out of range in DisplaySetRes (resmode)

Error Resolution: Value must range between 1 and 5

Error Example Fix:

```
Private Sub Command1_Click( )
Quirk1.ShowMsg("The resolution will change to 800 x 600")
Quirk1.DisplaySetRes(4)
' The Value must range between 1-5, in this case we used 4
```

-end of quirk.doc-