

In 3.2

New Features

Font Rendering -- text now drawn smooth and anti-aliased on Mac OS X (CoreGraphics font rendering) almost everywhere. Remaining exceptions include single-line EditFields, and a few listboxes in the IDE.

IDE Improvements

The following parts of the IDE use REALbasic's listbox to provide smooth anti-aliased text on Mac OS X; AERecord dialog, Capture Event dialog, CFM Plugin Editor, Code Editor's Code Browser, Control Order dialog, Database's Schema Editor and New Table dialog, Debugger's Stack Window, File Types Window, New Binding Dialog, Online Reference, Project Window, TabPanel Editor,

The following parts of the IDE have adjusted font and size to improve legibility; About Box, Build Dialog, Code Editor's Code Browser, Find Dialog, Language Reference, Menu Editor, Properties Window & Variables Window

The following parts of the IDE have adjusted layouts for Mac OS X or localization; Control Order dialog, Table Schema Editor and Field Options dialogs, Edit Tabs Dialog, Save

Code Editor -- Code Browser background of area around browser now white instead of grey on Mac OS X.

Properties Window, Variables Window, Window Editor Now much faster when dealing with medium (1,000) to very large strings (30,000+). Strings which have 5000 or more lines will no longer overwrite the viewing area of the window or control with screen trash.

Optimizations

Graphics.DrawPicture speed and memory efficiency improved. Avoids unnecessary caching in the case of drawing a picture created with the NewPicture method.

Miscellaneous changes and improvements

BalloonHelp and HelpTag -- wrong help tag or balloon help text no longer displayed, or to not be displayed at all. Affects windows both in the IDE and in built Mac applications; does not affect menu balloon help.

Carbon Resources

applications built for Carbon no longer contain 'PICT' resources 132 and 133 (used to fake Appearance controls on non-Appearance machines), nor 'carb' 0 (which is no longer needed).

applications built for Carbon now include a 128 pixel icon in the resource fork as 'icns' #128. You may override this by providing your own 'icns' #128 in any resource file in the project.

CodeEditor -- AutoCompletion's scrollbar length of the popup list is now as long as the popup window itself. In 3.1 it was sometimes 11 pixels short of that length.

Compiling -- Run or Make no longer proceeds very slowly (in some circumstances) except when the mouse was moving.

Databases

4D plug-in

SQL UPDATE command called through SQLExecute is more stable now. It no longer releases memory it shouldn't be touching in the first place.

no longer crashes when inserting a new record on WIndows.

OpenBase plug-in

_timestamp and _version fields are now hidden from .fieldschema because they are not eligible for changing. They can still be queried against.

closing the cursor no longer causes possible memory corruption problems. This could have resulted in your program crashing.

reading/writing blob data no longer corrupts database.

REALbasic database

comparison operator '>=' is no longer treated as a '<=' operation when comparing strings.

SQL UPDATE command called through SQLExecute is more stable now. It no longer releases memory it shouldn't be touching in the first place.

ErrorCode property is now initialized to 0, instead of some random number.

reading blobs (or long varbinary) works better on Windows.

database.ErrorMessage -- when inserting a new record, if the column name is unknown, REALbasic will now report the name of the invalid column, instead of just reporting "Unknown column".

Schema Editor

error message now reported if the database does not support dropping tables.

"Drop Table" button is now properly aligned to the right edge of the window when the window is resized.

type of each field and whether it is nullable shows (again).

Debugger

Array Viewer

Arrays of Objects Popup menu and title are now complete the first time you open them and REALbasic doesn't have to have an instance object created and inserted before it fills these values in.

large arrays (over 5000 elements) no longer cause array indexes to draw over each other. REALbasic can now easily display arrays much larger than this.

Medium (1,000) to very large arrays (30,000+) are now faster to update and scroll.

Resizing the ArrayViewer window no longer causes the scrollbar to loose sync with the contents of the viewer.

Clean-up -- leaked objects (which errant REALbasic programs may have left behind) releasing improved; it's now able to clean up leaked controls, for example. This clean-up happens at the start of each run, or when closing the project.

Stack Window -- Close Box, Grow Icon, and Zoom Box re-enabled in this window. They were turned off in the 3.0 alphas.

Listbox --

background pattern of other controls (e.g. the Canvas) no longer disrupted by the listbox.

DragItem in DragRow now has an appropriate drag rectangle. This fixed a problem of where classes were placed offscreen when dragging them from the Project Window to the Window Editor.

Hierarchical listboxes now hilight better on Mac OS X.

hierarchical listboxes now use a white background as the Mac OS X finder does.

resizing such that the scrollbar value changes draws the list display correctly.

ScrollBars are much smarter about how they adjust when the listbox content changes. The content no longer scrolls down automatically when rows are deleted or hidden, unless enough

rows are hidden or deleted so that the ListBox content would otherwise be blank. This is especially useful when implementing a hierarchical ListBox, since a row won't jump out from underneath a user's mouse after collapsing an item near the bottom of the list.

Scrollbars for Mac OS X moved one pixel to the right (one pixel down for horizontal Scrollbars). The default scrollbar size on Mac OS X is 1 pixel thinner; we were leaving 1 pixel of ListBox content to the right (or the bottom) of the ListBox.

Mac OS X

Double-Byte Character Systems strings in Japanese and other DBCS now drawn properly.

PopupMenu -- adding items no longer leaks memory.

Windows (the rectangular kind)

Floating Windows -- float above document windows under Carbon.

Window.FullScreen resizing a window to full screen mode no longer leaks memory.

Windows (the operating system)

Control.Refresh and RefreshRect -- controls now properly update when calling refresh.

ClearFocus now works on WIndows.

Destructors -- class destructors now called when quitting your application.

FolderItem.selectFolder -- dialog is now modal.

LittleArrows -- delay slightly increased between the first up/down event firing, and the next, for people with lazy hands.

MDI Child Windows

 minimize/maximize (i.e. zoom icons) boxes no longer on by default.

 setting the width and height of an MDI child window in code no longer moves it.

MouseDown event fires immediately, instead of waiting 20 msec before.

OpenPrinter -- width and height properties are more accurately reported when returning a graphics object.

Known Issues

Code Editor -- Code Browser Pane flickers on refresh on Mac OS X and when code entered for the first time

Databases -- REALbasic database BLOBs do not work on Windows. Writing them can corrupt the database.

Listboxes -- occasionally draws beneath it's bounds.

Tips

Extra Resources -- PICT resources 132 and 133 can be safely removed from your PPC built applications *if* you are certain that it will never be run on Macs which lack the Appearance Manager.

In 3.1

IDE Improvements

Code Editor Improvements

New: Autocompletion -- scrollbar appears when the list of completions is too long to fit on the screen.

Change: error messages are now positioned right under the line where the error has occurred, instead some distance lower. Also, the source code is scrolled all the way to the left when an error is displayed, so that the buggy line is sure to be visible.

Bug Fixes: Autocompletion:

now sports several enhancements and bug fixes: now completes "To", "DownTo", and "Step" in a For statement; now completes project items such as sounds, scripts, movies, etc.; now completes the properties and methods of project items; now completes the properties and methods of objects declared in any module (not just the first); and now completes the names of plug-in controls where appropriate

pop-up no longer moves the cursor when cancelled with the left-arrow or Delete key.

array and array element methods/properties now correctly complete.

menu bar no longer hides the list of solutions on systems with multiple monitors.

Scrollbars only appear on autocomplete pop-ups when actually needed.

typing the word "Menu" in the source code editor no longer causes the IDE to crash (especially on Mac OS X).

underscores in property or method names no longer confuses Autocomplete.

clicks past the end of a line no longer leaves the text cursor in an unusable state on Mac OS X.

inserting code into a long line under Mac OS X no longer causes refresh problems.

error messages (small yellow window) no longer have a title bar under some versions of Mac OS X.

Debugger:

New Feature: Break -- Control-C now invokes the debugger. When stuck in a long or infinite loop in the IDE, you can now break into the debugger by pressing the keyboard shortcut. Note that this also works in the classic IDE, but the old key sequence (command-shift-period) does not work under Carbon. The Carbon IDE in "Classic" supports no keyboard shortcut for "break" at all, though clicking in an open window of the IDE will usually "break". Note: Command-shift-period is now deprecated, and will be removed in a future release of REALbasic.

Bug Fixes:

Clean-Up -- viewing objects that were leaked (due to circular references), and then subsequently Running or quitting the IDE now no longer crashes.

Object Viewer and Array Viewers now properly draw scrollbars on Mac OS X, even when closed and then re-opened.

Opening an Object Viewer no longer causes REALbasic to "unexpectedly quit" under OS X.

Object Viewer, Array Viewer: scrollbar is now active when appropriate the first time the viewer is opened, without resizing the window.

Variables window: improved appearance in Mac OS X.

FYI: Command-shift-period to break into the debugger is now deprecated, and will be removed in a future release of REALbasic.

Known Issue: Break -- Carbon IDE in "Classic" supports no keyboard shortcut for "break" at all, though clicking in an open window of the IDE will usually "break".

Dialogs: buttons now expand or shrink to fit their captions, to more comfortably fit in localized versions of REALbasic. Among the dialogs updated are: About dialog; Preferences; File-Build Application; File-Print; Edit-FileTypes; Edit-Project Settings; Format-Control Order; Edit Constant

Value; and quit confirmation dialog

Menu Editor

New Feature: now has a more Aqua-like look when running the Carbon IDE under Mac OS X.

Bug Fixes:

- background of menu editor looks much better in 8-bit.

- cosmetic problems which left stray pixels in a few places fixed.

Menus:

New Features:

- Language Reference located in the Help menu rather than the Windows menu.

- Window, Minimize Window command -- this new command collapses a the active window in the IDE (equivalent to clicking the WindowShade or yellow-stoplight widget).

- Window, Bring All To Front command -- this new command brings all IDE windows to the front, as per the standard work-around for Mac OS X's new window layering behavior.

Online Language Reference:

New Feature: page titles now wrap to fit the width of the window.

Bug Fixes:

- canceling the autocompletion pop-up list no longer crashes.

- cosmetic problems when searching for "CDbl" (and a few other keywords).

- dragging source code snippets to the Code Editor now includes all the code in the snippet (previously only the visible portion would appear).

- finds words containing numbers (e.g., Open4DDatabaseByTCPIP).

- line breaks within tables (e.g., "CellType" in the Listbox class) now work.

Change: font size increased when running under Mac OS X.

modal dialogs for properties, methods, menu handlers, and constants no longer cause floating windows (toolbar, properties window, color palette) to re-appear.

Project File Reading:

New Features:

- IDE is now a bit more robust to certain types of file corruption in standard-format project files.

- IDE will now give you a warning (and then proceed, as always, to salvage as much of the data as it can) when a file appears to be corrupted.

Project items:

- cursor files can now be imported.

- dragging project items to a Finder window now exports the project item on Mac OS X.

- AppleScripts (in the project) now supported in the Carbon IDE and in Carbon builds.

Project Saving: saving over an existing project on a UFS volume no longer causes an error on Mac OS X.

Properties Window:

Bug fix: converting strings to integers now stops at the first garbage character.

Changes:

active area of "Visible" and "Private" checkboxes has been reduced to better fit the text, and to avoid cosmetic problems under Mac OS X.

heights of properties slightly changed to accommodate changes Apple has made after Mac OS X public beta.

Property Errors: Change: when a Run or Make finds an error in a property definition, the code editor for the object or class is now opened, and the property dialog appears with the error message contained therein, instead of in a separate little dialog. This fixes a modal dialog problem in Mac OS X, and saves a keystroke on all platforms. This change also applies to errors in method or event handler declarations.

Toolbar Bug Fixes:

- disclosure triangle and pop-up arrow icons no longer reversed in the OS X toolbar.

- hidden toolbars stay hidden after a run.

- fixed a bug which caused REALbasic to "Unexpectedly Quit" when dragging an item from the toolbar under certain versions of Mac OS X.

Database/Database Plug-ins Bug fixes

"Carbonized" plug-ins: CSV, DBF, OpenBase, Oracle 7.3, and PostgreSQL

OpenBase plug-in supports Windows.

4D Plug-in:

New Features:

- now supports BLOBs.

- added support for Windows.

Change for the Mac OS: Text fields are now reported as type 15 (longtext), instead of type 5 (varchar). Also, the reported length has been changed from 0 to 32767.

Known Issue (which we cannot fix): creating tables is not supported by 4D open (4D's server).

Bug Fixes:

- adding/inserting new records no longer leaks memory.

- Cursor.Update no longer unlocks the record being edited.

- error codes and messages are now populated if an error occurs, added more error checks.

- records are no longer locked unless the Edit method of a DatabaseCursor is called.

- ALTER TABLE SQL command -- the use of the 'COLUMN' identifier after ADD is now optional.

On the Mac OS:

- Fixed: /writing Dates reading/writing Times /writing Real Numbers reading/writing Booleans (accepts "True", "False", "T", "F", 1-9, 0 and slight variations thereof, including lowercase) /writing Pictures /writing 32k Text for updating a cursor and adding records to the database. Also fixed various crashing bugs related to writing to the database.

- 'OR' operator now functions correctly.

- time and boolean queries now work.

General:

New Features:

- queries now support the use of double quotes (single quotes are still valid as well).

Change: getstring returns the date in SQL date format which is YYYY-MM-DD, and HH:MM:SS for SQL time when used to get the string value of a date column in a cursor.

Bug Fixes:

DatabaseQuery -- memory leak when binding a databasequery to a listbox fixed.

doubles are now stored with maximum precision and are no longer truncated to 7 digits.

OpenBase plug-in built applications now work on Carbon.

On Windows:

Database.DoubleValue bad values no longer returned every other instance when retrieving a double value.

DataQuery binding: postgresSQL and openbase database querying problem fixed.

REALbasic database:

Doubles are now stored with maximum precision and are no longer truncated to 7 digits.

fixed problem with retrieving the length field from the field schema of a table. Although the results are always 0, it no longer crashes when you try retrieving the field.

'order by' clause now sorts float, double and date fields.

ALTER TABLE SQL command -- the use of the 'COLUMN' identifier after ADD is now optional.

On Windows:

keyword 'MAX' in a SQL query returns the correct result, instead of some seemingly random number.

'order by' clause no longer crashes when no records are found.

sum, min, max, and avg now work properly on Windows (byte order problem, again).

updating a row in a table that has a double column now works properly, whether or not you modify the double column.

Schema Editor:

New Features:

added button to drop a table.

field type pop-up now includes "long varbinary" among the pre-set options.

supported types/names now appear in the popup menu.

Bug Fixes:

"Indexes" button disabled when the database type does not support indexes.

long varbinary and timestamp now usable.

repositions controls when resized.

adding new columns for Openbase and PostgreSQL databases no longer causes problems.

PostgreSQL:

retrieving the date, time and timestamp fields no longer causes a crash.

Edit Fields

New Features:

EditField.Paste: this method pastes the (styled) text from the Clipboard to the EditField.

EditField.Copy: this method copies the selected text from the EditField, to the Clipboard,

preserving the style data as well.

Bug Fixes:

MultiLine editfields now shut down properly in Carbon. This may fix a memory leak.

fixed refresh problem which caused the editfield to update on top of a control, such as a tabcontrol, even if the editfield is in a hidden tabpanel or not visible.

now refrains from causing a redraw of controls above it when text or other attributes are changed, if it is not visible.

fixed a bug which caused single-line styled EditFields to appear in bold italic outline style on Mac OS X. This bug also afflicted the IDE itself, e.g., in the "New Method" dialog.

backspace key is now passed to your KeyDown event on Windows.

deleting characters from a password EditField, now actually deletes them instead of just erasing them on Windows.

fixed flicker problem that made the EditField redraw every time a key was pressed on Windows.

tabbing into a single line EditField now highlights the entire text on Windows.

EditField.KeyDown: tab keys now received, whether or not AcceptTabs is true on Windows.

EditField.TextStyleData: now saves with correct RGB color on Windows.

Listboxes:

Bug Fixes:

doubleclick event no longer fires when the mouse was clicked from one row to another really fast.

doubleclick event no longer fires when clicking off an item and on again rapidly.

extending a selection by pressing Shift-Down Arrow now works.

no longer possible to set the column count property in the properties window to more than 64 or less than 1. Doing so previously would cause a failed assertion (since we currently don't support more columns), trying to do so now will set the value to the maximum or minimum value.

column count changes no longer cause blank headers in listboxes on the Mac OS.

columns no longer disappear when the ColumnWidths string contains spaces on the Mac OS.

scrolling under Mac OS X no longer has cosmetic problems on Mac OS X.

restoring windows no longer causes headers to disappear upon restore (i.e. minimized to taskbar then restored to original form) on Windows.

scrollbars on a listbox no longer receive focus when tabbing into a listbox. The listbox now receives the focus on Windows.

setting the columntype of a column (other than 0) and adding a row no longer crashes on Windows.

tabbing into a Listbox with scrollbars from an Editfield the second time no longer crashes (because scrollbars no longer get the focus) on Windows.

ListBox.ColumnAlignment: decimal alignment now uses the decimal character of the user's local settings, instead of the default period character on Windows.

Menus

Bug Fixes:

Help menu now works in the Carbon IDE and in Carbon built applications.

QuitMenuItem appears in the application menu in built applications running under Mac OS X.

Edit Menuitems -- intrinsic menuitems now update when changes are made to the Editfield, so that the user doesn't need to select the menu for these items to enable/disable internally on Windows.

Miscellaneous changes and improvements

New Features:

Mac Specific: Sonnet G3 upgrade cards (in the "NumToString" toolbox call if you're curious) now work better with REALbasic. This bug caused a number of problems, including project corruption and invalid XML export files.

Carbon Specific: plist Resource: Carbon applications now include an automatically generated 'plist' resource which describes your application to the Mac OS X Finder. (If you wish, you may override this by including your own 'plist' 0 resource in a resource file in the project.)

Constants: dialog now remembers the data type selected the last time this dialog was used (very handy when entering a set of numeric constants, for example).

Date.TotalSeconds: now supports a greater range, corresponding to a greater range of dates.

Edit Value dialog: buttons now resize to fit their captions.

Mac OS X Specific

Sheets: some modal dialogs in the IDE are now sheets when running under Mac OS X. These include: New/Edit Property, New/Edit Method, New Menu Handler, and New/Edit Constant.

Error Messages: size adjusted to accommodate slightly larger text under Mac OS X.

Windows Specific:

Controls -- you can now use the arrow keys to navigate between controls, just like tabbing. The 'up' and 'left' arrow keys act like SHIFT+TAB, in that the focus shifts to the previous control in the control order, while the 'down' and 'right' arrow keys act like TAB, shifting the focus to the next control in the order chain. Some controls, such as EditFields, ListBoxes, ComboBoxes cannot be navigated through using the arrow keys, since these are used by the control themselves. In which case, you'll have to TAB out of them.

Control.HelpTags: multiline help tags now supported on Windows.

FolderItem.OpenAsSound: now recognizes MIDI, MP3, and MPEG files as valid sound formats. Note: to play MP3 files on Windows 95, check to see that you have the latest Windows Media Player installed. If not, go download it from Microsoft.com

System.Pixel: this new System method (Pixel(X as Integer, Y as Integer) as Color) returns the color of a screen pixel, specified in global coordinates. If the specified coordinates are not on any screen, then the color returned is black.

Optimizations:

Event Handling: a new optimization fixes sluggish behavior of EditFields and other controls which some users experience in projects run within the IDE.

Reading files is now faster, especially when backing up or checking for end-of-file frequently.

Opening Projects: reading a version 3 or XML format file is now faster -- in the case of a large XML file, it may be over 300% faster.

Bug fixes:

About Box -- multiple instances no longer possible

Carbon Specific:

AppleEvents:

AE Recorder: Captured Event window of the AE Recorder now works in Carbon.

Carbon built applications running under Mac OS 8 or 9 via CarbonLib now properly handle AppleEvents (such as 'quit')

CarbonLib: CarbonLib1.2.5 is what REALbasic uses. We recommend that you upgrade your copy of CarbonLib to match.

CarbonLib: version check -- both the Carbon IDE and Carbon built applications now check the version of CarbonLib they're running under when launched. If less than CarbonLib 1.1, they display an informative error dialog and quit, rather than crashing later.

Clipboard: fixed several bugs in the Clipboard class which could cause your application to unexpectedly quit.

Exception Handling: exceptions can now be caught in Carbon applications (or in the Carbon IDE).

Inline Input: single-line edit fields (including those in the IDE) now support inline Text Services Manager input (e.g., for typing Japanese) under Carbon.

Text Encoding Converters: supported for Carbon.

Compiler:

Missing Runtime Engine Function -- AppleScripts, StyledTextPrinter, or XCMDs in the Carbon IDE no longer cause this error to appear when building a project.

syntax errors now cause compiling to stop, this fixes some crashes and other odd behavior. This bug was introduced during 3.0 development.

Windows compiles after the first one no longer different from the first time. Also fixed problem that crashed the IDE when attempting to build for Win32 after the message "... a used plug-in does not support Win32 builds" appears.

Windows Builds: fixed problem with building inconsistent Windows executables when internal (BevelButton, ChasingArrow, etc.) or external (Valentina, etc.) plug-ins are used. Since memory was not being locked, the plug-in data may have been corrupted when building the executable.

Controls:

Carbon Specific

LittleArrows: Up and Down events now fire as they should under Carbon.

NotePlayer: supported on Carbon

Pictures: Carbon version of the IDE no longer produces invalid builds of projects that contain non-PICT images (e.g., JPEG pictures).

Mac OS X Specific:

enable/disable and activate/deactivate occurs in a more proper, Appearance-savvy way. Especially noticeable in scrollbars, which did not deactivate even close to correctly under Mac OS X.

BevelButtons: enabling via code when initially disabled in Mac OS X now works.

Canvas: Canvas.MouseDrag now flushes the graphics port (updates the screen) after calling any user code in this event on Mac OS X.

PushButton.Push: visible effect present on Mac OS X.

Windows Specific:

BevelButtons: transparent icons in bevelbuttons problem fixed on Windows.

BevelButtons: cosmetic changes on Windows.

DisclosureTriangle: cosmetic changes on Windows.

DisclosureTriangle: action event no longer fires if disabled on Windows.

CheckBoxes: SetFocus works on Windows.

ImageWell: fixed a resource leak when an ImageWell was refreshed.

ImageWell: fixed problem with drawing a red border instead of a grey border around the ImageWell.

PushButtons: SetFocus works on Windows.

RadioButtons: SetFocus works on Windows.

Scrollbars: enabled property respected when set in the properties window. Previously, always appear enabled when loaded, regardless of the "enabled" checkbox setting.

Exception Handling: exception objects no longer leak.

Focus Rings: now completely erased when a control became invisible (e.g., due to a tab panel switch).

Folder Items:

GetFolderItem: fixed memory leak when using GetFolderItem on Windows.

GetFolderItem: fixed problem with getting network paths on Windows.

GetOpenFolderItem: filter parameter is no longer ignored for Carbon.

SelectFolder: built applications under Mac OS 8.1 or older no longer experience a failure using SelectFolder.

Graphics Class:

Graphics.Pixel: now does bounds checking (both when getting and setting), thus avoiding crashes which were previously possible. Out-of-bounds accesses raise an OutOfBounds exception.

Graphics.Pixel: added work-around for bug in Mac OS X which caused assignments to a graphics Pixel to have no apparent effect.

Graphics.StringWidth: worked around an Apple bug which could cause an application to unexpectedly quit when getting the width of a string containing low ASCII values.

Help Menu: Language Reference item no longer appears in the Help menu of other applications while the REALbasic IDE (Classic) is running.

License Agreement: we now make sure to read the text from REALbasic's own resource fork, so it's not overridden by some naughty plug-in that doesn't adhere to the plug-in resource ID rules.

Mac OS X Specific

Background Pattern: draws correctly around controls drawn after some other controls (e.g. the TabPanel) on Mac OS X.

Carriage Returns: (line breaks) no longer appear as ampersands under Mac OS X.

ChasingArrows: visible property works in Mac OS X.

Class Instantiation: dragging a class to a window no longer crashes on Mac OS X.

Declares no longer fail in the IDE under Mac OS X.

Drag & Drop: built applications no longer crash when receiving a drop on Mac OS X.

GroupBox: caption changes no longer causes a GroupBox to erase its contents on Mac OS

X.

REALbasic icons: REALbasic now sports Big Beautiful Icons under Mac OS X.

Mouse drag behavior: MouseDrag events are generated even when the mouse isn't moving -- now more closely matches Classic and Win32 behavior. Also applies to some other situations, e.g., the automatic scrolling behavior in a listbox.

MouseDrag: the unwanted spinning-CD cursor during mouse drags has been eliminated.

Preferences menu item: disabled during a run on Mac OS X.

ProgressBar: when hidden (by visible=false or on a non-selected tab panel), progress bars now refrain from drawing anyway under Mac OS X.

Quit: hang no longer occurs when quitting the IDE under Mac OS X.

Quit handling: window CancelClose and application Close events are now properly received by applications run in the IDE under Mac OS X.

Refresh: Window and Control Refresh: now flushes the graphics port (forcing the screen to update).

RGBSurface.Pixel: fixed a bug which caused the Red component of the returned color to have a bogus value under Mac OS X.

Sheets: deactivating and then re-activating sheets via a mouse click no longer causes a crash under Mac OS X.

Single-Line EditField: Cut, Copy, and Paste now supported under Mac OS X.

Small dialogs: buttons are now resized to fit captions even under Aqua (e.g., in the auto-restore dialog).

Open Project dialog: Show Preview button and file type pop-up menu removed (did nothing, anyway).

Quitting:

fixed a bug which caused applications to crash (in the IDE or as built applications) when "Quit" was called from the action event of a bevel button.

fixed a bug which caused built applications to crash if a window's close event called "Quit", and some other code (e.g. in a button Action event) closed the window.

fixed a bug which could sometimes cause a crash when running again or quitting the IDE, after a run that leaked a control.

RectControl.Refresh: now has no effect when an item is invisible (either because visible=false, or because the control is on a hidden part of a tab panel).

Threads: multiple threads now more stable. Fixed a bug which could cause assertion failures, freezes, or crashes under certain circumstances in programs that had more than one Thread running at the same time.

Windows (the rectangular kind):

Floating Window:

created floating windows now no longer treated as a child Window on Windows.

floating Windows no longer disappear behind regular document Windows on Windows.

MDI applications no longer hide floating windows.

Global Floating Window:

activate properly when opened under Carbon.

EditFields and ListBoxes no longer lose the focus when clicking in the menu bar (or most anywhere else), after opening a Global Floating Window on the Mac.

fixed a bug which would have caused these to crash under Mac OS 8.1.

front most non-floating window no longer deactivates on Mac OS X.

bringing windows to the front is more reliable, e.g., when you summon the Window Editor from its Code Editor by pressing Option-Tab it comes to the front.

Background Color appears correctly when a window is first opened under Mac OS X.

Background Color stripes around controls in a window with a background color, when run within the IDE removed.

Window.Backdrop: white portions should now be transparent both in the IDE and at Runtime when transparent color is set to white. .

fixed problem with moving and sizing a Window in their Open event on Windows.

fixed problem with the minimum and maximum restricted sizes of a Window on Windows. When you resize the Window, the restricted size should prevent the Window from shrinking too small, or growing too large. It is now more accurate in calculating the restricted size, instead of being off a few pixels.

Windows OS Specific:

Cloning controls: fixed problem with cloning invisible controls at runtime, and they can now display their HelpTags.

Date: totalseconds now returns the correct result on Windows.

Date: assign to the month, day, and year in any order on Windows without worrying about whether the intermediate state is a valid date, now functions more like the Mac date class.

Events: MouseDown -- fixed problem with MouseUp event not firing if the MouseDown event had a Drag operation or a Contextual Menu popup on Windows.

Format: rounds properly on Windows, as the Mac does, when formatting a number in scientific notation. For example: Format(145678.5, "#.##e+") returns 1.46e+05 on Windows instead of 1.45e+05, as it did previously.

Icons: no longer have black and white reversed when the Win32 built application is created on a Mac running MacOS 9.1

Icons: 16x16 icons now supported (32x32 already supported), Previously the 16x16 versions of the icon were just reduced versions of the 32x32 ones.

Icons: 8-bit (256 colors) icons now supported, (previously supported 16 colors/4-bit) To take advantage of this make sure your project defines the 'icl8', 'ics8', 'ICN#' and 'ics#' resources, or paste in resources from another source that has these defined.

MDI Menubar: fixed bug where the MDI menubar would mysteriously attach itself to a floating or global floating window when setting the MenuBarVisible property.

MouseCursor: fixed problem with displaying MouseCursors. They were either showing up when they shouldn't have, or causing a lot of mouse cursor flickering.

MouseDown event: fires on Windows when the mouse button is down, even if no mouse movement is performed (same behavior of the Mac). This also fixes the problem with holding down the mouse button over the little arrows control, it should now fire the Up/Down event without having to hold down the mouse button and moving the mouse to instantiate the event.

PopupMenu: (a.k.a ComboBox on Windows) items are no longer truncated when you view them in the dropdown list. The dropdown menu should expand to accommodate the largest item, as the Mac does.

Picture.Draw: transparent images refresh properly when drawing (black refresh bug).

Plug-in Support: fixed the following bug -- under Win32, if a plug-in class "B" derived from another class "A" in the same plug-in, B's data would overlap A's data (causing a real mess).

Plug-in Support: the order in which Win32 plug-ins are loaded has been reversed to better match the loading order on the Mac. Note: try not to rely on this loading order anyway.

Shell: (NT only) new error code -3 and error message "Cannot open temporary file" is returned if a problem arises with reading the temporary output file.

Shell.Execute: now sets ErrorCode to -1 if execution fails.

Sliders: SetFocus works and controls can now be "tabbed into".

Socket.Listen: arrays of sockets now support listem. Previously this only worked for one socket in an array.

Sound.IsPlaying: now implemented.

String & Object parameters: fixed memory leak in passing a string or object as a parameter.

StyledTextPrinter: styled property of an EditField no longer needs to be enabled.

Tab Control: refresh problem with editfields on tabpanels, (although the fix extends to other controls as well), the editfield was the most noticeable.

Timer: fixed problem with timers firing too soon when mode was set to 1 in the IDE.

Timer.Mode: reset to 0 when timer with mode 1 fires.

Known Issues:

TabPanel: still doesn't line up its background with the window background properly on Mac OS X.

Text Encoding Converters: unsupported for 68k.

Movies:

New Feature: EditableMovie.Handle -- this property returns a handle to the editable movie, mainly used for toolbox calls that require the handle to a movie.

Change: Movie.Handle -- new Handle property deprecates the existing MovieHandle property. This new Handle property provides identical functionality as that of MovieHandle.

Bug Fixes:

movies in the Project window now correctly load and display in the Carbon version of the IDE .

MoviePlayer cleans itself up properly (when destroyed) in Carbon.

EditableMovie.Picture -- now returns a picture of the movie at the current position on Windows.

EditableMovie.Poster -- now returns a poster frame of the movie on Windows.

QuickTime:

Bug Fixes:

QTEffectSequence -- nil pictures no longer cause a crash when building a new effect sequence.

properties which now work under Carbon: Handle, MovieHandle, MovieHeight, BaseMovieHeight, BaseMovieWidth, MovieWidth, OpenURLMovie, MovieController, ControllerHeight, ControllerWidth.

Mac OS X now supported. For an overview of what these are: open the Language Reference, Click on the Theme Header and choose QuickTime.

QTEffect and QTEffectSequence now work under Win32

QTGraphicsExporter -- now works properly on Windows.

QTGraphicsExporter.SavePicture -- memory leak fixed on Windows.

QTVideoTrack.AppendPicture -- now actually appends a picture instead of doing absolutely nothing before on Windows.

Resources:

New Feature: file type of "rsrc" now recognized as a resource file, rather than requiring that the file be named "resources". You can also have more than one resource file in your project; the resources from all your resource files will be copied into the built Mac application. (In the case of a conflict, later resource files overwrite earlier ones, where the files are written in the order in which they appear in the Project window.)

Bug Fix: resource conflicts resolved by using later resources (now overwrite previous resources rather than causing a build failure) while building an application. Resources are written in this order: 1. plug-in resources (both code, and exported resources) 2. standard built-in resources (error dialogs, icons, etc.) 3. user's resource files (e.g. the traditional "resources" file) Note that the user's resources can overwrite the others.

Shared Libraries:

Bug Fixes:

fixed a bug which required you to open each shared library entry point in the entry point editor, for projects saved in standard (REALbasic 3) format.

shared library entry points no longer fail under some conditions in Mac OS X (fixed a compiler bug).

SpriteSurface:

Bug Fixes:

sprite image no longer causes screen trash when resized to something smaller.

fixed problem of not refreshing the screen during a Run, caused by the double-buffered graphics system in Mac OS X.

controls on the same window as a SpriteSurface now draw with a correct background color.

Optimization: slightly improved speed via an internal optimization.

Tips:

adding icons to your built application: put your icon (including 'icns' 32-bit icons) data in your "resources" file. These will now overwrite the icon resources supplied by REALbasic itself.