

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.107]

FB 5.7.108 requires OS X 10.6 or higher

[01] Lots of CocoaUI additions and changes - see "CocoaUI changes" below for details

[02] New CocoaUI demos for

GCD dispatch

TextView Attributed strings

TitlebarAccessoryView

DragDrop View

Several CocoaUI demos received bug fixes and updates to reflect current CocoaUI syntax.

[03] FB I/O keyword code (i.e. OPEN, READ/WRITE, LOF, RECORD, APPEND etc.) updated

It should deliver more consistent cooperative advisory locking on local storage. Network advisory locking not supported.

Watch for a separate list post.

[04] FBtoC crashes instead of reporting a Bad Factor on 'end fn = r' where 'r' is an undefined record name.

[05] FBtoC's Make Xcode Project(with stripping) now works.

[06] A unix function changed its parameter order. FBtoC code adjusted to conform.

CocoaUI changes

Application

new:

AppSetOutputWindow(SInt32 tag)

Box

new:

BoxContentViewMargins(SInt32 tag) = CGSize

BoxSetContentViewMargins(SInt32 tag, CGSize size)

ColorWell

new:

ColorWellSetColor(SInt32 tag, CGColorRef col)

Control

new:

ControlSetFormat(SInt32 tag, CFStringRef characters, Boolean pass, UInt32 maxLength, Boolean caseSensitive)

DialogEvent

new:

DialogEventArray = CFArrayRef

Menu

new:

MenuItemSetAlternate(SInt32 menuIndex, SInt32 itemIndex, Boolean flag)

MenuAddItem(SInt32 menuIndex, CFStringRef title, CFStringRef keyEquivalent, NSEventModifierFlags keyEquivalentModifierMask)

MenuInsertItem(SInt32 menuIndex, SInt32 itemIndex, CFStringRef title, CFStringRef keyEquivalent, NSEventModifierFlags keyEquivalentModifierMask)

SplitView

fix:

Unable to modify embedded views in nib splitview.

new:

standard menu statements

filemenu, findmenu, formatmenu, fontmenu, textmenu, viewmenu, helpmenu

TableView

new:

TableViewSelectRow(SInt32 tag, SInt32 rowIndex)

Text

new:

TextSetString(SInt32 tag, CFStringRef string)

TextSelectedRange(SInt32 tag) = CFRange

TextSetSelectedRange(SInt32 tag, CFRange range)

TextField

new:

TextFieldSetAllowsUndo(SInt32 tag, Boolean flag)

changes:

TextFieldSetFormat removed - use ControlSetFormat instead.

fix:

Focus ring appeared in wrong place when field's parent view flipped.

TextView

new:

TextViewSetAllowsUndo(SInt32 tag, Boolean flag)

TextViewSetImportsGraphics(SInt32 tag, Boolean flag)

TextViewSetAllowsImageEditing(SInt32 tag, Boolean flag)

TextViewSetAutomaticLinkDetectionEnabled(SInt32 tag, Boolean flag)

TextViewToggleAutomaticLinkDetection(SInt32 tag)

TextViewSetDisplaysLinkToolTips(SInt32 tag, Boolean flag)

TextViewSetAutomaticQuoteSubstitutionEnabled(SInt32 tag, Boolean flag)

TextViewToggleAutomaticQuoteSubstitution(SInt32 tag)

changes:

Default AllowsUndo setting is now `_true`.

TextViewString and TextViewSetString removed - use TextString and TextSetString instead.

View

new:

ViewAutoreizesSubviews(SInt32 tag) = Boolean

ViewSetAutoreizesSubviews(SInt32 tag, Boolean flag)

ViewRegisterForDraggedFiles(SInt32 tag)

dialog constants

`_viewDraggingEntered`

`_viewDraggingExited`

`_viewPerformDragOperation`

Window

new:

WindowViewWithTag(SInt32 wndTag, SInt32 viewTag) = ptr

WindowSetAutorecalculatesKeyViewLoop(SInt32 tag, Boolean flag)

WindowRecalculateKeyViewLoop(SInt32 tag)

WindowRecalculateKeyViewLoopByTagOrder(SInt32 tag)

WindowFirstResponder(SInt32 tag) = SInt32

WindowInitialFirstResponder(SInt32 tag) = SInt32

WindowSetInitialFirstResponder(SInt32 wndTag, SInt32 responderTag)

WindowContentRectForFrameRect(SInt32 tag, CGRect r) = CGRect

WindowFrameRectForContentRect(SInt32 tag, CGRect r) = CGRect

WindowSheetParent(SInt32 tag) = SInt32// macOS 10.9+

WindowSetResizeIncrements(SInt32 tag, CGSize size)

WindowSetContentSize(SInt32 tag, CGSize size)

WindowSetContentResizeIncrements(SInt32 tag, CGSize size)

WindowOrderRelativeTo(SInt32 wndTag, NSWindowOrderingMode order, SInt32 otherWndTag)

WindowSetLevel(SInt32 tag, NSWindowLevel level)

WindowStandardWindowButton(SInt32 tag, NSWindowButton btn) = ptr

WindowSetTitleVisibility(SInt32 tag, NSWindowTitleVisibility visibility)

WindowResizeFlags(SInt32 tag) = NSEventModifierFlags

20171030 FB 5.7.107

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.106]

FB 5.7.107 requires OS X 10.6 or higher

General Fixes & Additions

[01] Util_Array.incl's ArrayFirstObject() now works with 10.6+
[02] Editor crash opening a file when using _kFBProjTypeAlt1 = "_FB^π" because this constant translates to zero in C.
[03] New 'Abstract' demo and ViewSetNeedsDisplay demo added to CocoaUI Demos/Graphics.

[04] A later(Brian 20171101) supplement to 5.7.107 fixed a bad structure bug that causes FBtoC to quit. Issue happens when user fails to complete a block (like 'end select'). Code fix by Michelle.

CocoaUI changes 20171030(addendum to .107 by adding to .107's bundle)

Box

new:

BoxSetBorderRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)
BoxSetFillRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)
BoxSetTitleFontWithName(SInt32 tag, CFStringRef fontName, CGFloat size)

Button

new:

ButtonSetTitleRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)
ButtonSetBackgroundRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)

CocoaAlert

changes:

Previously, a sheet alert's tag value and return code would appear in dialog(-1), dialog(ev) respectively. They're now the other way around: returnCode = dialog(ev), alertTag = dialog(-1).

ColorPanel

new:

ColorPanelSetRGB(CGFloat r, CGFloat g, CGFloat b, CGFloat a)

ColorWell

new:

ColorWellSetRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)

Control

new:

dialog event constants - these are same values as _textFieldDidBeginEditing, etc

_controlTextDidBeginEditing

_controlTextDidChange

_controlTextDidEndEditing

ControlSetDoubleValue(SInt32 tag, double value)

ControlSetIntegerValue(SInt32 tag, SInt32 value)

ControlSetStringValue(SInt32 tag, CFStringRef string)

changes:

ControlSetActionCallback. The order of the callback function parameters has changed:

was: MyControlCallback(ctrlTag, wndTag, event)

now: MyControlCallback(event, ctrlTag, wndTag)

DatePicker

new:

DatePickerSetTextRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)

DatePickerSetBackgroundRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)

Menu

new:

MenuNumberOfItems(SInt32 menuIndex) = SInt32

PopUpButton

new:

PopUpButtonSetTitle(SInt32 tag, CFStringRef title)

PopUpButtonSetPreferredEdge(SInt32 tag, CGRectEdge edge)

PopUpButtonSetAutoenablesItems(SInt32 tag, Boolean flag)

PopUpButtonItemSetEnabled(SInt32 tag, SInt32 index, Boolean flag)

TextField

new:

TextFieldSetTextRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)

TextFieldSetBackgroundRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)

TextView

new:

TextViewSetBackgroundRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)

View

new:

ViewScrollRect(SInt32 tag, CGRect r, CGSize size)

ViewSetNeedsDisplayInRect(SInt32 tag, CGRect r)

ViewFrameRotation(SInt32 tag) = CGFloat

ViewBoundsRotation(SInt32 tag) = CGFloat

ViewsEmbedInView(SInt32 tag, ...)

ViewSetAcceptsFirstResponder(SInt32 tag, Boolean flag)

dialog events:

_viewKeyDown

_viewKeyUp

_viewFlagsChanged

Window

new:

WindowSetBackgroundRGB(SInt32 tag, CGFloat r, CGFloat g, CGFloat b, CGFloat a)

WindowSubclassContentView(SInt32 tag)

CocoaUI Changes/Additions(for .107)

Application

new:

AppSetTimer(CFTimeInterval interval, ptr callback, Boolean repeats)

Button

new:

ButtonSetImageScaling(SInt32 tag, NSImageScaling scaling)

ButtonSetBackgroundColor(SInt32 tag, CGColorRef)// macOS 10.8+

ButtonSetImageDimsWhenDisabled(SInt32 tag, Boolean flag)

change:

A new button given a title of "OK" (caps) is made the default (blue) button and responds to the return key.

A new button given a title of "Cancel" (case-sensitive) is made the cancel button and responds to the esc key and cmd-<full stop>.

CocoaAlert

new function and statement:

cocoaalert(tag, style, msg, info, buttonTitles) = SInt32 // currently requires Cocoa runtime (Cocoalnit)

cocoaalert tag, style, msg, info, buttonTitles, sheetFlag

Control

new:

ControlSetAction(SInt32 tag, CFStringRef actionName)
ControlSetActionCallback(SInt32 tag, ptr callback)
ControlSetEnabled(SInt32 tag, Boolean flag)

changes:

ControlSetFontWithName - if name is null or zero-length, previous font is used. If size == 0, previous size is used.

DatePicker

new:

DatePickerSetBezeled(SInt32 tag, Boolean flag)
DatePickerSetBordered(SInt32 tag, Boolean flag)

Menu

new:

windowmenu statement - puts up default Window menu in menu bar.

MenuItemSetAction(SInt32 menuIndex, SInt32 itemIndex, CFStringRef actionName)
MenuBarSetVisible(Boolean flag)
MenuBarHeight = CGFloat
MenuRemoveItem(SInt32 menuIndex, SInt32 itemIndex)
MenuRemoveAllItems(SInt32 menuIndex)
MenuItemWithTitle(SInt32 menuIndex, CFStringRef title) = SInt32
MenuPopUp(SInt32 menuIndex, SInt32 itemIndex, CGPoint location, SInt32 viewTag) = Boolean
MenuSetShowsStateColumn(SInt32 menuIndex, Boolean flag)
MenuSetFont(SInt32 menuIndex, CTFontRef font)
MenuSetFontWithName(SInt32 menuIndex, CFStringRef name, CGFloat size)
MenuItemSetState(SInt32 menuIndex, SInt32 itemIndex, NSControlStateValue state)
MenuSetValidateItemsCallback(ptr callback)

changes:

The cocoa menu statement no longer needs '-1' item index to refer to the menu itself - simply omit the item index param from the statement. e.g. cocoa menu 1,,, @"File"
Menus with index value > 100 are not added to the menu bar but stored elsewhere for later use (submenus, etc).

fix:

MenuSetSubmenu now works

NibMenu

change:

A menu assigned a tag value > 100 is given that same value to its index (see Menu changes above)

Panel

change:

options param removed from statement. Options are now set by function calls (see below)

new:

PanelSetFloatingPanel(SInt32 tag, Boolean flag)
PanelSetBecomesKeyOnlyIfNeeded(SInt32 tag, Boolean flag)
PanelSetWorksWhenModal(SInt32 tag, Boolean flag)

Popover

new:

dialog events
_popoverDetachableWindow
_popoverShouldClose
_popoverWillShow
_popoverDidShow
_popoverWillClose
_popoverDidClose
_popoverDidDetach// macOS 10.10
_popoverShouldDetach// macOS 10.10

PopoverSetDetachableWindow(SInt32 popoverTag, SInt32 wndTag)

PopUpButton

new:

PopUpButtonSetMenu(SInt32 tag, SInt32 menuIndex)

ProgressIndicator

change:

style param removed from statement. Style is now set by function calls (see below)

new:

ProgressIndicatorSetStyle(SInt32 tag, UInt32 style)
ProgressIndicatorSetIndeterminate(SInt32 tag, Boolean flag)

ScrollView

change:

options parameter removed from statement. Options are now set by functions (see below)

new:

ScrollViewSetHasHorizontalScroller(SInt32 tag, Boolean flag)
ScrollViewSetHasVerticalScroller(SInt32 tag, Boolean flag)
ScrollViewSetAutohidesScrollers(SInt32 tag, Boolean flag)
ScrollViewFlashScrollers(SInt32 tag)

SearchField

change:

options parameter removed. Options are now set by functions (see below)

new:

SearchFieldSetSendsSearchStringImmediately(SInt32 tag, Boolean flag)// macOS 10.10
SearchFieldSetSendsWholeSearchString(SInt32 tag, Boolean flag)// macOS 10.10
SearchFieldSetCentersPlaceholder(SInt32 tag, Boolean flag)// macOS 10.11
SearchFieldSetMaximumRecents(SInt32 tag, SInt32 value)// macOS 10.10
SearchFieldSetRecentsAutosaveName(SInt32 tag, CFStringRef name)
SearchFieldSetSearchMenuTemplate(SInt32 tag, SInt32 menuIndex)// macOS 10.10

SegmentedControl

change:

style and mode parameters removed from statement. Style and mode are now set by functions (see below)

new:

SegmentedControlSetMenu(SInt32 tag, SInt32 segmentIndex, SInt32 menuIndex)
SegmentedControlSetStyle(SInt32 tag, NSSegmentStyle style)
SegmentedControlSetTrackingMode(SInt32 tag, NSSegmentSwitchTracking mode)

Slider

change:

style parameter removed from statement. Style now set by functions (see below + ControlSetContinuous)

new:

SliderSetType(SInt32 tag, NSSliderType type)
SliderSetAllowsTickMarkValuesOnly(SInt32 tag, Boolean flag)
SliderSetTickMarkPosition(SInt32 tag, NSTickMarkPosition position)

Stepper

change:

valueWraps value defaults to _false

TableView

new:

now issues dialog event _btnClick
_tableViewDoubleClick dialog event
_tableViewSelectionDidChange dialog event

TableViewData(SInt32 tag) = CFMutableArrayRef
TableViewReloadData(SInt32 tag)
TableViewScrollRowToVisible(SInt32 tag, SInt32 row)
TableViewScrollColumnToVisible(SInt32 tag, SInt32 col)
TableViewSelectedRowIndexes(SInt32 tag) = CFArrayRef
TableViewNumberOfSelectedRows(SInt32 tag) = SInt32
TableViewIsRowSelected(SInt32 tag, SInt32 row) = Boolean
TableViewClickedRow(SInt32 tag) = SInt32
TableViewClickedColumn(SInt32 tag) = SInt32

changes:

Now allows unlimited number of columns
Editable columns supported
TableViewAddRow function removed

TextField

new:

TextFieldSetAllowsEditingTextAttributes(SInt32 tag, Boolean flag)
TextFieldSetImportsGraphics(SInt32 tag, Boolean flag)

changes:

TextFieldStringValue function removed - use ControlStringValue instead

TextView

new:

TextViewPerformFindPanelAction(SInt32 tag, NSTextFinderAction action)

Toolbar

change:

displayMode and options parameters removed from statement. These are now set by functions (see below)

new:

ToolbarSetDisplayMode(SInt32 tag, NSToolbarDisplayMode mode)
ToolbarSetAllowsUserCustomization(SInt32 tag, Boolean flag)
ToolbarSetShowsBaselineSeparator(SInt32 tag, Boolean flag)
ToolbarSetAutosavesConfiguration(SInt32 tag, Boolean flag)
ToolbarSetAllowsExtensionItems(SInt32 tag, Boolean flag)

ToolbarItem

new:

ToolbarItemSetToolTip(SInt32 toolbarTag, SInt32 itemTag, CFStringRef toolTip)

View

new:

ViewsHidden(SInt32 tag) = Boolean
ViewsHiddenOrHasHiddenAncestor(SInt32 tag) = Boolean
ViewNextKeyView(SInt32 tag) = SInt32
ViewNextValidKeyView(SInt32 tag) = SInt32
ViewPreviousKeyView(SInt32 tag) = SInt32
ViewPreviousValidKeyView(SInt32 tag) = SInt32

Window

new:

WindowSetDelegateCallback(SInt32 tag, ptr callback)
WindowAddChildWindow(SInt32 parTag, SInt32 childTag, NSWindowOrderingMode ordered)
WindowRemoveChildWindow(SInt32 parTag, SInt32 childTag)
WindowParentWindow(SInt32 tag) = SInt32
WindowSetHidesOnDeactivate(SInt32 tag, Boolean flag)
WindowSetHasShadow(SInt32 tag, Boolean flag)
WindowSetFrameOrigin(SInt32 tag, CGPoint pt)
WindowSetFrame(SInt32 tag, CGRect frame)
WindowCascadeTopLeftFromPoint(SInt32 tag, CGPoint pt)
WindowsZoomed(SInt32 tag) = Boolean
WindowPerformZoom(SInt32 tag)
WindowZoom(SInt32 tag)
WindowSetShowsResizeIndicator(SInt32 tag, Boolean flag)
WindowSetMaxFullScreenContentSize(SInt32 tag, CGSize size)// macOS 10.11+
WindowSetMinFullScreenContentSize(SInt32 tag, CGSize size)// macOS 10.11+
WindowOrderBack(SInt32 tag)
WindowsVisible(SInt32 tag) = Boolean
WindowRemoveFrameUsingName(CFStringRef name)
WindowSetExcludedFromWindowsMenu(SInt32 tag, Boolean flag)

20171005 FB 5.7.106

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.105]

FB 5.7.106 requires OS X 10.6 or higher

General Fixes & Additions

- [01] Programmatically creating toolbars with Source-file combining set to OFF had a bug.
- [02] TWM (The Window Maker) updated to version 1.9.15
- [03] Apple macro name changed causing crash in FileHandling.c (__Require_noErr is new name)
- [04] CFIndex.incl header and its runtime file index.c now 64-bit compatible
- [05] New Util_Sound.incl header and examples.
- [06] Get/Put/Kill Preferences statements now are 64-bit.
Note: this work did NOT involve the headers Util_CFPrefs.incl and Util_CFPrefsCFStrings.incl.
- [07] Util_Array.incl updated with two new convenience accessors, ArrayFirstObject() and ArrayLastObject()
- [08] Quartz Cocoa.bas demo (for Quartz 1.15 in FB Examples/Graphics/CoreGraphics(Quartz)) updated to use GraphicsCurrentCGContext and ViewBounds().
- [09] FB's file I/O OPEN statement does *not* honor POSIX advisory locking when data files are located on a server (either macOS Server or macOS File Sharing). This means OPEN "N" and OPEN "R" will not return appropriate errors if the file is already locked by another process. Paolo C. suggested a runtime change which works in his environment (see "File Bug in 5.7.105" list thread in late September 2017 for details) but it isn't a general solution. FBers needing to access server files should consider other file access methods other than OPEN if their files are on servers.

CocoaUI Changes/Additions

Application

New:

_appOpenURLs event constant

AppEventString = CFStringRef
AppEventArray = CFArrayRef
AppEventDictionary = CFDictionaryRef
AppEventURL = CFURLRef

Changes:

_appOpenFile and _appOpenFiles constants removed. Use _appOpenURLs instead.

ColorPanel

New:

ColorPanelColor = CGColorRef
ColorPanelSetColor(CGColorRef col)
ColorPanelSetMode(NSColorPanelMode mode)
ColorPanelSetShowsAlpha(Boolean flag)
ColorPanelAlpha = CGFloat
ColorPanelShow
ColorPanelClose

Control

New:

ControlStringValue(SInt32 tag) = CFStringRef
ControlTakeDoubleValueFrom(SInt32 tag, SInt32 fromViewTag)
ControlTakeIntegerValueFrom(SInt32 tag, SInt32 fromViewTag)
ControlsEnabled(SInt32 tag) = Boolean
ControlSizeThatFits(SInt32 tag, CGSize size) = CGSize // macOS 10.10
ControlSetContinuous(SInt32 tag, Boolean flag)
ControlPerformClick(SInt32 tag)
ControlSetLineBreakMode(SInt32 tag, NSLineBreakMode mode)

Dialog

Changes:

DialogEventCGContext removed. Use GraphicsCurrentCGContext instead.

DialogEventRect no longer returns the view rect, and now returns its dirtyRect. To get the view rect, use either ViewFrame or ViewBounds, whichever is appropriate.

Event

New:

EventLocationInWindow = CGPoint
EventModifierFlags = NSEventModifierFlags
EventTimestamp = CFTimeInterval
EventType = NSEventType
EventWindow = ptr
EventKeyRepeatDelay = CFTimeInterval
EventKeyRepeatInterval = CFTimeInterval
EventCharacters = CFStringRef
EventCharactersIgnoringModifiers = CFStringRef
EventsARRepeat = Boolean
EventKeyCode = unsigned short
EventPressedMouseButtons = UInt32
EventDoubleClickInterval = CFTimeInterval
EventMouseLocation = CGPoint
EventClickCount = SInt32
EventLocationInView(SInt32 tag) = CGPoint

Geometry

Geometry.incl header removed. That header contained NSRectEdge constants NSMinXEdge, NSMinYEdge, NSMaxXEdge & NSMaxYEdge. Use: CGRectXxxx. instead.

Graphics

New:

GraphicsCurrentCGContext = CGContextRef

ImageView

Syntax change:

options param removed. Attributes are now set with functions (see below).

New:

ImageViewSetAnimates(SInt32 tag, Boolean flag)
ImageViewSetEditable(SInt32 tag, Boolean flag)
ImageViewSetAllowsCutCopyPaste(SInt32 tag, Boolean flag)

LevelIndicator

Syntax change:

options param removed. Attributes are now set with functions (see below).

New:

LevelIndicatorSetEditable(SInt32 tag, Boolean flag)
LevelIndicatorSetTickMarkPosition(SInt32 tag, NSTickMarkPosition position)

Menu

Change:

If tag values are given to menus and items, those values are now returned to FB's on menu function instead of their index values.

New:

MenuSetTag(SInt32 menuIndex, SInt32 itemIndex, SInt32 tag)
MenuSetAutoenablesItems(SInt32 menuIndex, Boolean flag)

SegmentedControl

Fixed: Crash when setting segment image. This crash only occurred on machines running macOS 10.12 or earlier and when multiple segmented controls in a window.

Stepper

Syntax change:

options param removed. These attributes are now set with functions (see below).

StepperSetAutorepeat(SInt32 tag, Boolean flag)
StepperSetValueWraps(SInt32 tag, Boolean flag)
StepperSetContinuous(SInt32 tag, Boolean flag)

TextField

Syntax changes:

behavior param removed. These attributes are now set with functions (see below).

lineBreakMode param removed. This attribute is now set with a function (see Control.incl).

New:

TextFieldSetEditable(SInt32 tag, Boolean flag)
TextFieldSetSelectable(SInt32 tag, Boolean flag)

TextLabel

Syntax change:

lineBreakMode param removed. This attribute is now set with a function (see Control.incl).

View

Syntax change:

subclass param added (default = `_false`). Enabling subclass allows the user to pick up `_viewDrawRect` plus other dialog events (see below).

New:

ViewAddSubviewPositioned(SInt32 superviewTag, SInt32 subviewTag, NSWindowOrderingMode position, SInt32 otherViewTag)
ViewSetWantsLayer(SInt32 tag, Boolean flag)
ViewSetFrame(SInt32 tag, CGRect r)
ViewSetFlipped(SInt32 tag, Boolean flag)

ViewAnimator(SInt32 tag) = ptr
ViewAnimatorSetFrame(SInt32 tag, CGRect r)
ViewAnimatorSetFrameOrigin(SInt32 tag, CGPoint origin)
ViewAnimatorSetFrameSize(SInt32 tag, CGSize size)
ViewAnimatorSetFrameRotation(SInt32 tag, CGFloat rot)
ViewAnimatorSetBounds(SInt32 tag, CGRect r)
ViewAnimatorSetBoundsOrigin(SInt32 tag, CGPoint origin)
ViewAnimatorSetBoundsSize(SInt32 tag, CGSize size)
ViewAnimatorSetBoundsRotation(SInt32 tag, CGFloat rot)
ViewAnimatorSetFrameCenterRotation(SInt32 tag, CGFloat rot)
ViewAnimatorSetAlphaValue(SInt32 tag, CGFloat value)

dialog events:
 _viewMouseDown
 _viewMouseDragged
 _viewMouseUp
 _viewMouseMoved
 _viewMouseEntered
 _viewMouseExited
 _viewRightMouseDown
 _viewRightMouseDragged
 _viewRightMouseUp

Window

New:
 _windowContentViewTag - this constant identifies a window's content view

WindowSetTitle(SInt32 tag, CFStringRef title) // this now works for both Carbon and Cocoa windows

Demos:

Couple of new graphic demos showing GraphicsCurrentCGContext() and ViewBounds() usage

FB Runtime Status

64-bit Runtime	Comments
AppThings.m	
CocoaUI.m	
Containers.c	
FileHandling.c	
FileHandlingUtils.m	
General.c	
Index.c	
OSPanel.m	
Prefs.c	
StringUtils.m	
SystemDirectoryCopyURL.m	
32-bit Runtime	Comments
AppThings.c	mostly supplanted by CocoaUI
ConsoleApp.c	conditionally included from FBtoCConsole.incl. No longer supported
EditFields.c	essentially replaced by CocoaUI - remains available for 32-bit builds
FilesDollarFunction.c	supplanted by OSPanel. Obsolete/removed in current FB builds
Util_FileDirectory.c	All functions except FD_PathCreateCFURL are 32-bit only

 20170919 FB 5.7.105

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.104]

FB 5.7.105 requires OS X 10.6 or higher

Introduction to New Containers

This release's container update could prevent your app from compiling and could cause it to malfunction. Continuing the on-going work to move the FB runtime to 64-bit, FB containers were updated to remove Handles and use pointers.

Because containers are now based on pointers, container usage relying directly on Handles needs updating to work correctly. Creating a container directly from a Handle won't work but dereferencing the Handle and using a helper function (see below) will.

- Creating/modifying containers is basically the same except for the following:
- Toolbox Handle-involved calls (Set/GetHandleSize, PtrToHand etc.) cannot be used to manipulate containers.
 - Reading/Writing to/from a container changes slightly because a container is a pointer
 - Passing container pointers to other functions changes slightly.
 - Code **CANNOT** dereference containers directly. The helper functions **MUST** be used (see 'To/From Pointer/Container' below)

- Container size is limited based on whether the app is 32-bit(approx. 2 GB) or 64-bit(> 2GB).
- 32-bit user interface widgets like edit fields might have size limitations
 - using 32-bit edit fields (or any other 32-bit UI widget) precludes a 64-bit app.

Containers and Edit fields. Unchanged

- Put container contents in an edit field. `edit$(_myEditField) = #gC`
- Load container from edit field. `gC = edit$(edit_field_id)`
- Load edit field from container. `edit$(id) = #gC`

To/From Pointer/Container. New

`p = fn ContainerToPointer(@gC) // get a pointer from a container.`
 Replaces handle syntax: 'myHandle = [@gC]'

`fn ContainerFromPointer(@gC, p, size) // load container from pointer.`
 Replaces handle syntax: 'gC = &h'

Both may be found in Util_Containers.incl

Old Handle syntax NOT supported

- Loading containers from handle data. Use `fn ContainerFromPointer` instead of `gC = &h`
- Loading a handle from a container. Use `fn ContainerToPointer` instead of `h = [@gC]`
- Adding TEXT resource data to a container doesn't work. There is no replacement. `gC = %128`
- Handle toolbox calls won't work on containers (i.e. DisposeHandle, NewHandle, GetHandleSize, SetHandleSize).

File I/O to/from a container

- NEW* - reading using FB file I/O

```

dim as pointer p
open "I", 1, @url
size = lof( 1, 1 )
p = fn malloc( size )
read file #1, p, size
fn ContainerFromPointer( @gC, p, size )
fn free( p )

```
- NEW* - writing using FB file I/O
// Create a pointer from a container and write the file from the pointer.

```

dim as pointer p
dim as container gC
open "O", 1, @url

```

```

size = len( gC )
p = fn ContainerToPointer( @gC )
write file #1, p, size
fn free( p )

```

3. New I/O functions from *Util_Containers.incl* don't require FB I/O statements

```

fn ContainerWriteToURL( @gC1, url ) // Write
fn ContainerReadFromURL( @gC2, url ) // Read

```

Note: "Util_Containers.incl" must be in the project if these new functions are used.

Most of the FB container statements and Util_Container.incl have the same usage even though internals have changed

Statements and functions with same usage are:
right\$\$, left\$\$, mid\$\$, len, asc, ucage\$\$, string\$\$, FBCompareContainers(),

Util_Containers.incl continues to have some functional overlap with FB keywords.
Those functions are listed below with FB equivalents

fn ContainerUppercase	ucage\$\$
fn ContainerInsertPascalString	mid\$\$ statement
fn ContainerFindPascalString	instr
fn ContainerAppendPascalString	gC += "some pascal string"
fn ContainerFindAndReplacePascalString	instr followed by mid\$\$ statement

CocoaUI Updates and Additions

Application

New:
AppKeyWindow
AppMainWindow
AppIsHidden
AppHide
AppUnhide
AppUnhideWithoutActivation
AppIsActive
AppActivateIgnoringOtherApps
AppHideOtherApplications
AppUnhideAllApplications
AppShowHelp

Box

New:
BoxSetTransparent
BoxSetTitleFont
BoxSizeToFit

Button

New:
ButtonSetTitleColor
ButtonSetAttributedTitle
ButtonSetAlternateTitle
ButtonSetPeriodicDelay
ButtonSetAttributedAlternateTitle
ButtonSetAlternateImage
ButtonSetTransparent
ButtonSetAllowsMixedState
ButtonSetNextState
ButtonHighlight

ColorWell

New:
ColorWellActivate
ColorWellDeactivate
ColorWellDrawWellInside

Control

New:
ControlSizeToFit
ControlSetFont
ControlSetFontWithName
ControlTakeStringValueFrom

DatePicker

New:
DatePickerSetBackgroundColor(
DatePickerSetTextColor(
DatePickerSetLocale
DatePickerTimeInterval
DatePickerSetTimeInterval
DatePickerMinDate
DatePickerMaxDate

Dialog

Changes:
1. Dialog events get/set functions consolidated, e.g. CocoaWindowEventXxxx, TabViewEventXxxx, SplitViewEventXXX, etc. are all now DialogEventXxxx.
2. on dialog fn now receives window content view _viewDrawRect event for CG drawing. Obtain the CGContextRef with DialogEventCGContext and content view rect with DialogEventRect.

CocoaUI Header files

Changes:
Removed 'Cocoa' prefix from filenames

ImageView

New:
ImageViewSetCGImage

LevelIndicator

New:
LevelIndicatorMinValue
LevelIndicatorMaxValue
LevelIndicatorWarningValue
LevelIndicatorCriticalValue
LevelIndicatorSetStyle

Changes:
1. Style param removed from LevelIndicator statement. Style is now set by LevelIndicatorSetStyle function.

Menu

Changes:
MenuItemEventSetBool renamed MenuEventSetBool

NibView

Changes:
viewIdentifier param now optional. If this param is omitted, the first NSView found in the nib will be used.

NibWindow

Changes:
wndIdentifier param now optional. If this param is omitted, the first NSWindow or NSPanel found in the nib will be used.

SecureTextField

New:
SecureTextFieldSetEchosBullets

SplitView

New:
SplitViewAdjustSubviews

Text

New:
TextString

TextField

New:
TextFieldSetTextColor
TextFieldSetBackgroundColor
TextFieldSetFormat

TextView

New:
TextViewSetBackgroundColor
TextViewSetDrawsBackground
TextViewSetUsesRuler
TextViewSetRulerVisible
TextViewSetUsesInspectorBar
TextViewSetSelectedRange
TextViewSetContinuousSpellCheckingEnabled
TextViewSetGrammarCheckingEnabled
TextViewStartSpeaking
TextViewStopSpeaking
TextViewSetUsesFontPanel
TextViewSetUsesFindPanel
TextViewSetUsesFindBar
TextViewTextStorage
TextViewSetRichText

View

New:
ViewWithTag - returns view ref for issuing direct obj-c calls
SInt32 ViewSuperview
Sint32 ViewWindow
void ViewSetFrameOrigin
void ViewSetFrameSize
void ViewSetFrameRotation
CGRect ViewBounds
void ViewSetBounds
void ViewSetBoundsOrigin
void ViewSetBoundsSize
void ViewSetBoundsRotation
CGFloat ViewAlphaValue
void ViewSetAlphaValue
CGFloat ViewFrameCenterRotation
void ViewSetFrameCenterRotation
void ViewPrint
void ViewSetNeedsDisplay
void ViewRotateByAngle

Window

New:
WindowWithTag - returns NSWindow ref which can be used for issuing direct obj-c calls
WindowSetAspectRatio
WindowSetContentAspectRatio
WindowSetDocumentEdited
WindowDocumentEdited
WindowToggleFullScreen
WindowSetAllowsToolTipsWhenApplicationIsInactive
WindowSetTitlebarAppearsTransparent
WindowSetTitleWithRepresentedFilename
WindowSetTitleWithRepresentedURL
WindowAttachedSheet

Changes:

Previously, only non-Cocoa apps would subclass window content views in order to draw focus rings around fields. Now, Cocoa apps also subclass the content view and intercept the drawRect method for CG drawing. Obviously, focus ring drawing is not applied to fields in Cocoa apps.

NibWindow

Changes:
A negative tag value in the nibwindow statement now works same as cocoa window statement (loads window invisibly).

General

Changes:
Functions requiring a higher macOS than set in Base SDK no longer silently fail and now report "implicit declaration of function 'xxxx'..."

Fixes & Additions

- [01] Added InstallCocoaUIScripts app to FB Help menu
- [02] Writing data in "R" mode could truncate file length (FH_WriteData truncateFileAtOffset:) - reported by B.G.
- [03] FBDelay() within General.c had a value conversion issue
- [04] FBLOf() within FileHandling.c. Added conditional compile for 32/64-bit to silence compiler warnings
- [05] PrintContainerWnd() in AppThings.c updated to use pointers instead of handles.
- [06] Util_String.incl updated to silence warning for containsString: method.
- [07] MidStmt_Ctnr() in containers.c was not doing the stack pop. reported by Mark.
- [08] New version of Quartz (1.14) and demos for use with CocoaUI or Carbon.
- [09] Editor source updated to use new pointer-based containers
- [10] "Unknown type name 'FBMenuEventInfo' when compiling -DDECARBONATE -m64 and CocoaUI.

20170625 FB 5.7.104

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.102]

FB 5.7.104 requires OS X 10.6 or higher

Summary

Note: This release contains one or two updates which may prevent your app from compiling.

Updates:

- Bug fixes.
- 64-bit updates.
- New project template updated to new window statement syntax.
- Run statement now accepts CFString path or CFURL. The old 'path\$' syntax is deprecated.
- Kill Field statement removed. Those needing to continue with this functionality should either use an older release or use the direct toolbox call: if (h) { if ((HGetState(h) & 0x20) == 0) DisposeHandle(h); }
- System() function updated. Many old selectors removed.
- New FB header, Util_RunningApplication.incl, provides information about running application(s). See FB Examples/CoreFoundation & Cocoa/Cocoa/Util_RunningApplication.incl example(s)/RunningAppDemo.bas.
- TWM updated to version 1.9.14.
- New FB Header, Util_UserDefaults.incl, is a functional replacement for Util_CFPrefs.incl.
- On a dialog _btnClick event, dialog(-1) now returns the button's window number.

More Details

- [01] Run statement updated to Run a CFString path or a CFURLRef. Old syntax deprecated. FBHelp updated too. Brian/Bernie
- [02] Delay statement updated to 64-bit. Uses nanosleep(). Brian
- [03] Kill Field removed for same reasons Read/Write Field were in FB 5.7.99.
Those needing to continue with this functionality should either use an older release or the direct toolbox call:
if (h) { if ((HGetState(h) & 0x20) == 0) DisposeHandle(h); }
- [04] System() function updated. Many old selectors removed. _aplActive updated to modern 64-bit code.
_scrnWidth, _scrnHeight, _sysVers, _crntDepth are the other remaining selectors. FB Help updated accordingly & FBtoC reports errors for any other selectors.
- [05] FBGetProcessInfo() is now conditionally compiled only in 32-bit compiles. This assures the old code won't be there for 64-bit compiles.
FBHelp updated to reflect #5 and #6
- [06] New FB Header, Util_RunningApplication.incl, provides information about running applications (duh!).
See FB Examples/CoreFoundation & Cocoa/Cocoa/Util_RunningApplications.incl example(s)/RunningAppsDemo.bas
- [07] FBKillPicture() for now is within #if !__LP64__ to make sure it is NOT included on 64-bit builds.
At some point all of the PICTURE commands need review. Either update to use a non-Quickdraw graphic or remove.
- [08] Bug fixes for AddKeywordToSymbolTable() and AddSubKeywordToSymbolTable() Symbol Table Implementation. Brian
- [09] TWM (The Window Maker) updated to version 1.9.14. Bernie
- [10] A new Util_UserDefaults.incl functionally replaces Util_CFPrefs.incl and is 64-bit. Note: the older Util_CFPrefs.incl has older Handle code in it. Bernie.
- [11] dialog(-1) now returns the window number on _btnClick events. Bernie
- [12] Various code cleanup in General.c. AppThing.c Brian
- [13] Some parenthesized coordinates in Window statement fail. Bernie
- [14] Window attributes fixes for new Window statement when attributes are missing or NULL. Bernie
- [15] New Project template updated to new Window statement syntax. Brian
- [16] Save panel 'Cancel' button not honored when saving a new editor window. Bernie
- [17] New reserved words introduced. Note that reserved words cannot be used as variable names and will fail when building with "Bad factor in expression in line xxxx". The most likely new words to cause a failure are: 'imageview', 'panel', 'scrollview', 'slider', 'toolbar' and 'view'. For more information on these new reserved words, see CocoaUI.pdf in FB's Help menu

Note: Containers are unchanged in this release but an upgrade effort to 64-bit is in progress. If raw pointer speed isn't required, a 13-July-2017 list post, "Feedback request FB's containers", suggests alternatives.

20170625 FB 5.7.102

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.101]

FB 5.7.102 requires OS X 10.6 or higher

New

- [01] Window & Appearance Window statements changes: Brian/Bernie
 - (a) **Appearance Window** statement is **deprecated** and the translator warns and suggests moving to the Window statement.
 - (1) Otherwise same as FB 5.7.101.
 - (b) **Window** statement's parameters are identical to Appearance Window except the rectangle and window attribute have changed.
 - (1) Rectangle parameter: If used, must be a 'CGRect'; a QuickDraw 'Rect' is no longer accepted. (x1,y1)-(x2-y2) format still available.
 - (2) Window attributes: Are specified in a standard array instead of OR'ing into a integer. Also, the attribute constant names have changed.
- [02] New Grand Central Dispatch (GCD) demo (at: FB_5_7_102_Examples/CoreFoundation & Cocoa/GrandCentralDispatch(GCD)/GCD Timer Demo)- Steve VV
- [03] OSPanelSave now defaults to hiding the file extension - Bernie
- [04] FB header, Tlbx CFString.incl, updated to include CFStringInitInlineBuffer and CFStringGetCharacterFromInlineBuffer which are designed for fast access to string characters. - Brian
- [05] New CoreText include with demo now in FB_5_7_102_Examples/Text/CoreText/CTTextFrame - Bernie
- [06] FB's "Printing Without a Window" feature now uses anti-aliased text for PRINTing. Implemented via QDSSwapTextFlags() and srcOr text transfer mode.
This was added as a convenience for those who need to run a quick test. It does not upgrade general window code. FBers are still responsible for windows they create.
- [07] FinderInfo example using CFURL added. See: FB_5_7_102_Examples/Files/Droplet apps/FinderInfo replacement(CFURL-based) - Bernie
- [08] An example showing how to configure an info.plist for app and document icons. See: FB_5_7_102_Examples/Miscellaneous/info_plist for doc & app icons - Bernie
Was discussed on the list on 8-June-2017 and 10-April-2017
- [09] New CFString and CFNumber utilities contained in two new headers (Util_String.incl and Util_Number.incl)
Provides the power of Cocoa/Objective-C methods but wrapped in easy-to-use FB functions.
Examples are in: FB_5_7_101_Examples/CoreFoundation & Cocoa/CoreFoundation/Strings and Numbers
Note: the content of the older Util_String(s).incl was combined with the new Util_String.incl

Fixes/Updates

- [10] Editor source files with forward '/' slashes in the names sliently fail to save file contents. Brian
Within the Editor's source code, OSPanel was substituted for files\$ and any file name slashes converted to ':'(as NSSavePanel does) as noted by Bernie and Ken
The issue arises due to the the Editor code's need to concatenate a parent and file name URL.
Once/If FSSpec usage is replaced with CFURLRefs in the editor's global array, URL concatenation and the scan to check for forward slash (/) will be unnecessary.
- [11] Tlbx MoreFilesX.incl (supports deprecated file calls and structures) has been removed from the Headers. All FBers should switch to more modern calls. Brian
- [12] ConsoleWindow support removed. Recommend using NSLog.
- [13] Launching the Editor by double-clicking a single source file sometimes fails if old TYPE/CREATOR (specifically TYPE) is zero
Editor was incorrectly interpreting such a file as a package project file, and then ReadProjectIntoDataBase() fails because it isn't a project. Brian
- [14] gNewWndPositionMethod global within Rntm Appearance.glbl was causing an "unknown type" error for '-m64 -DDECARBONATE' builds. Bernie.
Solution: Restore gNewWndPositionMethod's definition to within the '#if ndef __LP64' so it is only used in 32-bit builds.
- [15] Open "0" now defaults to OSTypes of zero for Type/Creator. Also see Def Open below.
- [16] Def Open now defaults to NULL OSTypes for Type and Creator and can be requested explicitly(e.g. Def Open "")
Def Open's FBHelp updated to reflect changes.
Note: Def Open is deprecated. Recommend not using it and instead specifying UTIs and file extensions for icons in the application's info.plist. Bernie/
Brian
Apple Documentation includes UTIs that have been standard since OS X 10.4:
https://developer.apple.com/library/content/documentation/Miscellaneous/Reference/UTIRef/Articles/System-DeclaredUniformTypeIdentifiers.html#//apple_ref/doc/uid/TP40009259-SW1

also: <http://tinyurl.com/jd7uj2l>
- [17] Updated FBHelp and FBtoC to reject Read/Write Field (per change in FB 5.7.99) a little more gracefully. Brian
- [18] FixSDK updated to version 1.2.5. Bernie
 - (a) Fixes a bug that caused some tableview buttons to vanish.
 - (b) Now can Export/Import SDKs to to/from an external directory(s). See FixSDK Help for more details.
 - (c) FixSDK now launches fully when the 10.6 SDK is not in Xcode (the previous version told the user to quit and install the 10.6 SDK).
If a 10.6 SDK is not installed, all buttons are disabled except for Import which allows the user to locate a 10.6 backup. Once installed, the buttons become enabled.
 - (d) Button enabling logic bug fix
 - (e) Alert sheets warn users that existing SDKs will not be overwritten
 - (f) 'Show SDKs in Finder' file menu item.
- [19] Search Apple Headers(SAH) app updated to version 1.0.5. The SDK popup menu showed '0.0' due to Apple's use of alises in their SDKs. Fixed. Bernie
- [20] TWM (The Window Maker) updated to version 1.9.13 supports the new Window Statement syntax. Bernie
- [21] NSLog.incl updates: settings window is a popover instead of drawer if available. Also implemented a bug fix- Bernie
- [22] Introduced as an option in release 5.7.42 (February 2016), the following statements now require CFString titles and no longer accept pascal strings
 - (a) Appearance Window
 - (b) Window
 - (c) Appearance Button
 - (d) Button
 - (e) apple menu (also accepts CFArray of titles)
 - (f) Edit Field
 - (g) Menu statement
 - (h) Shutdown statement

20170425 FB 5.7.101

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.100]

FB 5.7.101 requires OS X 10.6 or higher

New

- [01] Util_Workspace now a standard FB Header - Bernie
 - a) Access to many NSWorkspace options in simple FB functions
 - b) See the examples at FBExamples/Files/Util_Workspace examples

- [02] Util_Array.incl now a standard FB Header
 - a) Useful helper functions call to some of NSArray's methods. Bernie
 - b) One cool function is ArrayWithObjects() and could be used to pass UTIs to the new OSPanelOpen() as Bernie showed on list on 21-April-2017
 - c) See the examples at FBExamples/CoreFoundation & Cocoa
- [03] Util_Dictionary.incl now a standard FB Header
 - a) Useful helper functions call to some of NSDictionary's methods. Bernie
 - b) See the examples at FBExamples/CoreFoundation & Cocoa

Fixes/Updates

- [04] OSPanelOpen/OSPanelSave's FBHelp received some minor modifications to clarify the type of parameters it accepts. Brian (pushed out in later update to .100 too)
- [05] FB 5.7.100 issues files\$ deprecation warnings and suggests OSPanelOpen/OSPanelSave. Brian
- [06] Files\$'s FBHelp updated to issue deprecation warnings. Brian
- [07] Minor updates to FBtoCConsole.incl to use OSPanel instead of files\$. Bernie
- [08] Minor update to MDArray.incl in FBExamples. Bernie
- [09] FBtoC printed a double "Warning:". Bernie
- [10] Apple's new file alias inclusion in its SDKs generates a Base SDK popup with a nameless first item. Bernie.
- [11] Editor file save crashes when user selects Cancel. Bernie
- [12] FBtoC crashes trying to execute showAlert() and its subordinate [NSAlert runModal]. Needs to run on main thread. Brian
- [13] Editor updated to use OSPanel for File=>Open... menu option. Modernizes code and allows it to remember the last directory the FBer opened when used next time. Brian
- [14] Minor updates to OSPanel's FB Help to clarify allowFileTypes are file extensions and UTIs and not OSTypes
- [15] VAList in FBExamples updated to use the C macros.

 20170413 FB 5.7.100

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.99]

FB 5.7.100 requires OS X 10.6 or higher

New

- [01] OSPanelOpen/OSPanelSave are new built-in FB keywords. They offer more options than files\$ but don't use the Carbon framework. Bernie with some nudges from Brian
- [02] FBHelp updated to include OSPanelOpen/OSPanelSave. Bernie/Brian

Fixes

- [01] 'redefined variable' errors if both Util_FileDirectory.incl & Util_FileManager used in same project/file. Bernie
 - [02] DEFSTR long incorrectly generates '8' instead of '4', so any functions using the gPDefStrSize global receive the wrong size. Brian
 - [02] TWM - The Window Maker - builds graphical window and generates FB source - now included in FB5. Bernie
- ALSO: the following copy of the on-list announcement contains good information:

Hello Everyone,

This release introduces OSPanel and its new FB keywords, OSOpenPanel and OSSavePanel. Since they are keywords, there are no extra includes to remember and are part of the standard FB runtime. They provide the functional equivalent of the old files\$ plus additional features. Here are some (but not all) new features:

- Designate a starting directory for the panel(dialog)
- Remembering directory from last open
- Designate tags
- Extending a dialog with your own controls
- Sheet use¹

Additionally, the new keywords are very flexible (parentheses and all parameters optional). Plus there are other subordinate new keywords that can be used to prepare the panel's appearance and other related needs.

The best way to learn about all the new keywords is exercise the demos at 'FBExamples/Files/OSPanel examples' and review the FBHelp.

Recommendation:

OSPanel is now recommended instead of files\$ or the (carbon-based) NavDialog calls and both of the latter are deprecated and should not be used going forward. files\$ is available in 5.7.100

Reason for change:

Even though files\$ was limited to using CFURLRefs in the last release, it continued to use Carbon-based NavDialog calls for the dialog UI. The change to OSPanel makes all the code Carbon-free.

1. Sheets attached to FB-created (i.e. appearance window) parent windows may exhibit detachment of the parent window in some specific scenarios and other undesirable window drawing artifacts. Apparently, not every OS releases exhibits this problem but the issue is rooted in a parent Carbon window (i.e. parent Cocoa windows are fine). The demos show two ways of doing sheets both with/without Cocoa windows.

 20170320 FB 5.7.99

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.97]

FB 5.7.99 requires OS X 10.6 or higher

- [01] FileHandling.c updated to 64-bit - Brian
- All FB's I/O verbs no longer accept FSSpecs/FSRefs and only accept CFURLRefs. See Jan/Feb 2017 FB list discussions for details.
- a) the file-based version of the OPEN statement must specifically use the CFURLRef version (OPEN UNIX and OPEN "C" excepted because not file-based)
 - b) READ/WRITE FIELD are removed and no longer supported due to use of deprecated Handles. Replacement options were discussed in Jan/Feb 2017 FB list discussions.
 - c) FB Help's 'Appendix A - File Object Specifiers' updated to reflect removal of FSRef/FSSpec support and sample code updated.
 - d) FB's Rename verb now accepts only CFURLRefs. Syntax is: rename urlForCurrentFile urlForNewFile.
 - e) FB's Kill now accepts only a CFURLRef for the file name.
 - f) FB's files\$ now only uses/returns a CFURLRef for each of the three major options listed in FB Help. FB Help updated to reflect changes.
 - 1) "ConsoleWindow"'s usage of files\$ updated to use URL. I don't like it because ConsoleWindow depends on an FSSpec for its TXNSave() use.
 - 2) New constants for files\$ _URLOpen, _URLFolder and _URLSave are preferred over their predecessors _CFURLRefOpen, _CFURLRefFolder and _CFURLRefSave but have the same values.
 - 3) OPEN/FILES\$ changes to Editor source to allow it to build with FB 5.7.99.
 - 4) FBtoC flags non-CFURLRef files\$ usage for the variable but does not check the mode constant; but runtime honors only valid modes.
 - a) Make sure to check files\$'s returned fileName for non-zero length. Zero indicates failure and possible bad mode constant.
 - g) Apps crashing with a "Bad file descriptor" message indicates a file i/o attempt which isn't allowed by the OPEN mode (i.e. opening in mode "I" and trying to WRITE the file).
 - h) A second attempt (either within the app or another app) to OPEN "N" on a file currently/already open in "N" mode has specific new behavior compared to prior FB5 versions:
 - 1) The second attempt process/code is given read-only access (essentially "I" mode) to the file.
 - 2) The FB runtime sends a "file already open with with write permission" (_opWrErr) error code which can be used by the caller (must be trapped with 'on error' and "error")
 - 3) Even though opened for "I" after a request for OPEN "N", it must be closed or any subsequent exclusive OPEN requests will fail.
 - 4) FBers should always check the error status returned by any file OPEN and take appropriate steps.
 - 5) N.B. OPEN code only provides access; it does not provide any data integrity protection when there are concurrent readers and writers of the same file.

[02] Util_FileManager additions - Bernie

- a) File access functions written with only modern 64-bit code. Eventual replacement for Util_FileDirectory.
- b) Only supports CFURLRefs and does not support FSRefs or FSSpecs.
- c) Several demos available in the FB Examples
- d) Util_FileManager.incl includes Util_PathUtilities.incl and Util_URL.incl
- e) Util_FileDirectory is still available for those who need it but folks should plan to migrate to Util_FileManager.
- f) For fn FM_TrashItemAtURL (but not the other functions in the include). The FBer has two choices:
 - 1. Its current implementation uses a method requiring OS X 10.8+
 - 2. Another method, which is currently commented out and doesn't have the 10.8+ requirement, is possible but it is deprecated in OS X 10.11. Once you've decided which constraints are best for you, simply comment out the one you don't want and uncomment the other (or vice-versa). Consider using the user includes folder if maintaining it yourself.

[03] New/Changed Headers and Examples - Bernie, Steve VV, Brian

- a) A slick multi-dimensional Core Foundation array implementation handles all the details. See: FB_Examples/CoreFoundation and Cocoa/MDArray(multi-dimensional)
- b) Tlxb CFUUID.incl and Tlxb CFStringTokenizer.incl added to Headers

- c) CFMeasurement headers and demo added to FB Examples
- d) Superscript/Subscript demos added to FB Examples
- e) Util_Files.incl uses outdated (parameter block) file functions. N.B. IF YOUR CODE USES FUNCTIONS FROM THIS INCLUDE, IT WON'T BUILD.

[04] Default error function - Brian

(i.e. when it isn't supplied via 'on error fn yourErrorHandler' in your own code) updated to include:

- a) error string and comment
- b) N.B. Carbon calls GetMacOSStatusErrorString() and GetMacOSStatusCommentString() were NOT used. Cocoa's NSError used [NSError

errorWithDomain:NSOSStatusErrorDomain

[05] NSLog.incl updates - Bernie

- a) silence a clang 'dealloc' warning (accidentally omitted in 5.7.97)
- b) Copy & Copy All items added to text view's contextual menu
- c) NSLogBeginEditing & NSLogEndEditing. Bernie's explanation follows:

Multiple NSLog calls in big loops can make it appear that your app is hanging. After issuing NSLogBeginEditing, the NSLog text view is only updated at the point NSLogEndEditing is called.

```
// - Example 1 ---
```

```
On my machine, this takes about 95 seconds for the text to appear.
```

```
for i = 0 to 150000
  NSLog(@"string %d",i)
next i
```

```
// - Example 2 ---
```

```
Enclosing the code in an NSLogBeginEditing/NSLogEndEditing pair takes less than 4 seconds.
```

```
NSLogBeginEditing
```

```
for i = 0 to 150000
  NSLog(@"string %d",i)
next i
```

```
NSLogEndEditing
```

Fixes - Brian

[06] FBtoC Help, inadvertently omitted in 5.7.97, is back again.

[06] Editor scrolls source view to an incorrect line number but FBtoC's line number is accurate (captured value of global in local before doing a dispatch async)

[08] Comparing an FB container to a string constant resulted in string stack failure (changed gFBStk to SInt16 in Runtime.h and General.c so it can go to -1 instead of 65535)

[09] FBtoC's Settings dialog would not show up when requested via the Editor's preference pane option. Thanks to RC and BW for noting and reminding.

Internal: Added source for "Error Codes" app to FB project source

ALSO: the following copy of the on-list announcement contains good information:

Hello Everyone,

Please read the following introduction before downloading and replacing your current FB; it is important to know the impact. Download link at the bottom.

Introduction

Unlike some FB5 releases, this one contains several changes which impact your current FB source code. Depending on your code, it may not compile with this FB version.

Internally, all of FB runtime disk-based file I/O system (with some specific exceptions which will be noted later) was updated and is now 64-bit compatible. In some cases the updates impact the operation and syntax of specific FB file I/O keywords. Most FB file I/O keyword's operation and syntax are exactly the same while a few have slightly different syntax/requirements and behavior. It is important for FBers to understand these impacts, assess the work involved, and decide how/when to implement this new FB release. Like usual, old releases and the prior release (5.7.97) are available. Hopefully, this moves us closer to a "Future" BASIC. The design goal was to keep the language in familiar territory with the same keywords while upgrading the supporting runtime.

Overview of Changes to FB File I/O

As usual, the [Release History Notes](#) detail the changes but here is an overview.

[A] File-based versions of the OPEN statement(open modes "I", "O", "A", "R" and "N") must specifically use the CFURLRef version. OPEN no longer supports FSSpec or FSRef.

[B] OPEN "N" now provides an automatic read-only open when a file is already open in "N" mode.

It notifies the code of the automatic selection of "I" mode, when "N" was requested, via FB's standard ERROR functions. See code below at note:¹

[C] READ/WRITE FIELD are removed due to use of deprecated Handles and potential links/associations with old code.

Replacement options were discussed in Jan/Feb 2017 FB list discussions. See this link for a possible replacement strategy: <http://freegroups.net/groups/futurebasic/0:67289read.html>

[D] FB's FILES\$ now uses/returns only a CFURLRef for each of the three major options(i.e. selecting a file, a folder, or save location/name)

[E] Util_Files.incl uses outdated (parameter block) file functions. N.B. IF YOUR CODE USES FUNCTIONS FROM THIS INCLUDE, IT WON'T BUILD.

In all cases this include's functions can either be replaced with similar URL functionality or the functionality is obsolete (Resource fork stuff). If you can't find modern replacements, please ask on the list. One replacement example: fn FSSendFileToTrash(sourceObj as ^FSSpec) can be replaced with one of the new Util_FileManager.incl code. (see FB Examples==>Files==>Util_FileManager Demos==>TrashItemAtURL

[F] Coding errors can cause crashes because the error isn't found until it runs.

For example:

(1) A "Bad file descriptor" crash message indicates a file i/o attempt which isn't allowed by the OPEN mode (i.e. opening in mode "I" and trying to WRITE the file)

(2) Using the wrong fileId. i.e. opening file #1 for read and then trying to read file #2 when fileId #2 doesn't exist.

The [Release History Notes](#) detail many other additions and changes and FBers are strongly encouraged to read and understand their impact.

What didn't change in FB File I/O

[F] Even though their underlying code changed a lot, most FB file I/O verbs/keywords use the same syntax. For example, Error, Close, Write#, Input#, Print#, Record, LOF, EOF, Rec, Loc, Read File, Write File, Append, Open "C" and Open "UNIX" all have the same syntax. Such is the benefit of an abstracted high-level language.

[G] The relative performance and usage recommendations remain the same. For example, using read# with multiple variables generates more code(potentially a lot more) than doing a read file# into a FB record (C structure) or pointer. Read# generates one physical read per variable versus one read for the entire FB Record in Read File#. Here is an example with an FB record reading four variables with Read File vs. Read:

```
begin Record myRec
dim as short   a
dim as long    b
dim as Str255  c
dim as double  d
end Record
dim as myRec   aRec
```

```
This FB:      read file #2, @aRec, fileSize
Generates this C: FBReadFile( 2, (void*)(void*)&aRec, false, fileSize );
```

```
This FB:      read #2, aRec.a,aRec.b,aRec.c;13,aRec.d
Generates this C: aRec.a = FBReadSwapShort( 2 );
                  aRec.b = FBReadSwapLong( 2 );
                  FBReadString( 2, (char*)&aRec.c, 13 );
                  aRec.d = FBReadSwapDouble( 2 );
```

[H] This FB release brings lots of new headers, updates and bug fixes. Most timely is the new addition of Util_FileManger to assist with the new OPEN and FILES\$. Thank you Bernie.

Notes:

1. Code below shows interception of an OPEN "N" notification and how to determine if "I" mode was assigned because file is in use.

```
//-----  
include "NSlog.incl"  
  
dim as CFURLRef url  
dim as short ioError : ioError = 0  
dim as CFIndex fileSize  
dim as CFStringRef s  
dim as dim as p  
  
s = @"/Users/brians/Desktop/mytextfile2.txt"  
url = fn CFURLCreateWithFilePath( NULL, s, _kCFURLPOSIXPathStyle, _false )  
  
on error end  
error = _noErr  
open "N",1, @url  
ioError = error  
error = _noErr  
// _opWrErr ( -49 ) file already open with with write permission*/  
// in this case another process has the same file open in mode "N"( since this process does not ),  
// so the runtime opens the file for you in read-only mode ( like mode "I" ) and notifies  
// the FBer of this action by sending back the _opWrErr code. Brian 20170303  
  
select switch ( ioError )  
  
case _noErr  
    // No problem, your code has the file open in "N"  
  
case _opWrErr  
    // other code ( or code in another process ) already has the file open in "N"  
    // so your code now has this file open in "I" mode  
    NSLog(@"ioError = %d", ioError )  
    fileSize = lof(1,1)  
    p = fn malloc( fileSize )  
    read file #1, p, fileSize  
    s = fn CFStringCreateWithBytes( NULL, #p, fileSize, _kCFStringEncodingMacRoman, _false )  
    free( p )  
    NSLog (@"%s",s )  
    CFRelease( s )  
    Close #1  
end select  
  
CFRelease( url )  
  
RunApplicationEventLoop  
//-----
```

General note: Several FBers contribute to the update effort and the release notes now identify the persons mostly (approximately 95%+ of the effort) responsible for the coding effort in each area, even though others not mentioned might have contributed.

20161228 FB 5.7.97

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.94]

FB 5.7.97 requires OS X 10.6 or higher

[01] FBtoC is now a 64-bit app. This means we have a translator that will run even if Apple removes Carbon - Brian

Bonus: removal of more runtime code reduced FBtoC's disk size.

[a] Translate/Compile runs on a separate queue/thread(libdispatch driven) to reduce/eliminate FBtoC's previous 'beachball' cursor.

Note: Macs with fewer than two cores will not see all the benefits of concurrent multi-thread processing but the 'beachball' should be (mostly) gone.

[03] Remaining 'route_toBuffer' and associated Handle management code replaced with CFMutableStringRef(one for errors and one for code).

[04] NSTableView convenience functions added to Util_UI.incl header.

[05] FBtoC app is codesigned (FB editor is not codesigned) for Apple Gatekeeper compatibility.

[06] FBtoC now copies storyboard files created in Xcode into the app package. See Storyboard Demos in FB Examples/Cocoa.

[07] Minor update to NSLog.incl silences a clang 'dealloc' warning.

Fixes - Brian

[08] Long constants (> 255 characters) within BeginCxxxx/EndC would be flagged as errors (string too long).

[09] Multiple successive invalid ids passed to Edit\$ statement in user program overflows gFBStk's maximum value.

[10] fn AddConstantsFromResourcesToSymbolTable incorrectly converted old built-in constants to unsigned long instead of signed long. This resulted in "implicit conversion" compiler warnings and could cause bugs.

[11] Building an empty (or with one line like 'print') single source file over twenty times would crash. Forty successive tests succeeded without a crash.

Might be fixed.

[12] fn WriteCodeToURL would crash CFReleaseing a null CFURLRef. The problem is the path passed to WriteCodeToURL() was longer than the 1024 characters allowed, so the conversion to CFURLRef fails. Same potential issue fixed in fn WriteCharToURL too. N.B. note the path length limitation.

Coding and implementation: Bernie, Brian and Deep

2016xxxx FB 5.7.95

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.94]

FB 5.7.95 requires OS X 10.6 or higher

[01] FBtoC now copies storyboard files created in Xcode into the app package. See Storyboard Demos in FB Examples/Cocoa.

[02] Minor update to NSLog.incl silences a clang 'dealloc' warning. In Headers.

[03] Bug: fn WriteCodeToURL would crash CFReleaseing a null CFURLRef.

The problem is the path passed to WriteCodeToURL() was longer than the 1024 characters allowed, so the conversion to CFURLRef fails. Fixed Same potential issue fixed in fn WriteCharToURL too.

N.B. 5.7.94 and 5.7.95(internal) are the last FBtoC versions to be all FB source code. Most of 5.7.97 is FB source but it was converted to Xcode before 5.7.97 was released. Bottom line: if you want source for 5.7.97+, look at the Xcode source and NOT the FB source. Brian 20170111

Coding and implementation: Bernie, Brian and Deep

20160916 FB 5.7.94

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.69]

FB 5.7.94 requires OS X 10.6 or higher

[01] Updated fn GetNumber in Lexical Analysis - internally it's all CF but it still returns a pascal string. A start on the last conversion phase, lexical analysis.

[02] Remaining left\$, right\$, mid\$, ucase\$ and instr usage converted to CF. One 'hex\$' remains.

[03] Consolidated five functions (fn GetLeftParenthesis, right parens, brackets, equals, comma) into one function

[04] Updated GetLexeme_Priv in lexical analysis to CF and remove all the 'exit "GetLexeme_Priv"' so it will generate a normal C switch statement

[05] CFStringFromPStr() changed to accept pointer to Str255 instead of just Str255. Saves an extra copy for every call to it.

[06] Code buffers updated directly instead of using intermediate string. Translation speed improves 40 to 50% in some cases.

[07] New FixSDK app updated to handle Xcode 8 SDK changes. Thanks to Bernie for the research and updates.

Fixes

(08) An 'xref @' variable translates incorrectly. Fixed. (thank you to Paolo C. for reporting)

Coding and implementation: Bernie, Brian and Deep

20160815 FB 5.7.69

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.49]

FB 5.7.69 requires OS X 10.6 or higher

Fixes

- (01) After reporting user code errors, FBtoC would lock up and not finish. Fixed. (thank you to Peter B. for reporting)
- (02) Crash when launching FBtoC from desktop(i.e. without the Editor) with no build_goodies/Headers files. Fixed
- (03) HexString() incorrectly truncated when honoring some 'defstr' (i.e. byte, word, long) settings. Fixed.

FBtoC internal code updated to 64-bit

- (04) fn EmitData updated to use CF and remove FB's dynamic array
- (05) fn GotoStatement updated and its subordinate fn FixLabel (now named fn FixLabelCreateCF)
- (06) fn GsubStatement updated and its subordinate fn FixLabel (now named fn FixLabelCreateCF)
- (07) fn ReadQuotedString updated to CF and renamed fn ReadCreateQuotedString
- (08) Miscellaneous global pascal strings used in 'Exit' code changed to CF mutable strings - see fn ClearMiscGlobalStrings
- (09) Prefix labels for both generated and source (i.e. 'LL' and 'L') changed to CF - see kprefixForLabelFromFBSource/kprefixForGeneratedLabel
- (10) All 'Exit' related code updated to CF including fn UniqueLabel
- (11) Changes to fn BSCmparePStr() and fn BinarySearch() to use CFStrings. Superficial changes at the moment.
- (12) Many fn InString calls replaced with fn CFStringFind for slight performance boost.
- (13) Some Unix functions, such as fn CopyUnixCommandFirstLineResponse, updated to not use FB runtime
- (14) Replaced Handles with CFStrings for fn StringExpressionContinuation and other 'String Expression & Assignment' functions
- (15) FSp LongName File Utilities removed and its only active function, fn FSpTrashObject, moved to Cocoa File Utils.incl
- (16) SendTextFragmentToCommentBuffer() updated to Core Foundation
- (17) Various 'Copy Resources Phase' FNs changed to CFStringGetCharacterAtIndex() for small speed gains.
- (18) Prelexical State: PrelexicalState record changed to use CFStringRef instead of Str255. PushPrelexicalState(), PopPrelexicalState() changed to manage the CFStringRef memory appropriately.

Coding and implementation: Bernie, Brian and Deep

20160711 FB 5.7.49

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.48]

FB 5.7.49 requires OS X 10.6 or higher

- (1) Tlxb CFSet.incl, Tlxb CFBag.incl and Tlxb AppThings.incl added to Headers
- (2) CF replacement functions for right\$, left\$, mid\$, instr and hex\$ now easily accessible via new header file, Util_Strings.incl.
- (3) Crash converting old project file format to new. Fixed.
- (4) Error reporting fn was releasing the error string too early (fn BuildStringAndReportError) which could cause unexplained crashes. Fixed.
- (5) Build Settings: edit buttons for 'Path to Developer folder' and 'Compiler Options' were inoperative. Fixed.
- (6) Several translation bugs fixed related to 'def fn using', dynamic arrays and other issues. These were coding errors made while converting FBtoC source. Mark(LabHelper X) and Peter's(Drafting) source were instrumental in identifying issues.
- (7) Smart tabs: If the user enters an initial tab value on creation, it failed to select the tab correctly: Fixed.

appearance button _cTabs,, 2,,, @"Tab 1;Tab 2", @r, _kControlTabLargeNorthProc

Coding and implementation: Bernie, Brian and Deep

20160527 FB 5.7.48

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.47]

FB 5.7.48 requires OS X 10.6 or higher

Thanks to Bernie's prolific coding efforts, tab control and radio group implementation is totally easy. SmartTabs/SmartRadioGroups are implemented in the FB runtime, so all the messy work is done for you there.

If you'd like to try it, download the example at: FB_5_7_48_Examples/Controls/SmartTabs+ RadioGroup1

Important: Please Note (N.B.)

FB 5.7.42, in February 2016, announced the next FB release will not support pascal strings in appearance window, appearance button, apple menu and menu statements. That change is still planned but there is no firm schedule yet. Those statements continue to support both pascal and CF strings in 5.7.48 but FBers should be making changes if not done already.

Coding and implementation: Bernie, Brian and Deep

20160527 FB 5.7.47 rev 2

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.47]

FB 5.7.47 requires OS X 10.6 or higher

- (1) read # and write # were not translating correctly causing the build to fail with compile errors. Fixed.
- (2) Two 'Quit' items on FBtoC's menu negatively impacted operation. Fixed
- (3) Added CFRelease(msg) to STACK_PUSH() to quiet the analyze warning
- (4) Build Setting, 'Check Array Bounds', failed to generate compilable C code in some scenarios. Fixed.

Coding and implementation: Bernie, Brian and Deep

20160516 FB 5.7.47

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.46]

FB 5.7.47 requires OS X 10.6 or higher

- (1) New Stack_Push() function replaces older macro because clang was emitting hundreds of warnings about potential unsequenced errors. Function not only avoids any sequence errors but checks stack pointer to make sure it is within the stack array bounds. If not, it alerts the user and quits FBtoC.
- (2) FBtoC would spew errors after multiple(20+) builds of same project(see 5.7.45 notes #3). Stack pointer not being decremented in FBOpenUnix(). Fixed.

Coding and implementation: Bernie, Brian and Deep

20160515 FB 5.7.46 - development only

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.45]

FB 5.7.46 requires OS X 10.6 or higher

- (1) SmartTabs plus SmartRadioGroups - temporarily turned off until serious bug can be fixed.

Coding and implementation: Bernie, Brian and Deep

20160504 FB 5.7.45 - development only

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.44]

FB 5.7.45 requires OS X 10.6 or higher

- (1) Tlxb ControlDefinitions.incl updated to fix errors in GetDataBrowserUserState() and SetDataBrowserUserState()
- (2) Four code buffers(declarations, main, functions and comments) converted to CFArrays of CFMutableStrings (formerly arrays of allocated Handles).

(3) Crash building FBtoC source with "source combining" turned off resulted in crash when memmove() overwrote the end of the array and corrupted another variable.

Increasing gFBStrStk's size from 32 to 128 via kMaxStringStackLevel constant *bypasses* the bug. Bug was found/fixed in 5.7.47

(4) Edit Field's FB Help updated to reflect option to use Core Foundation string in title/text

Coding and implementation: Bernie, Brian and Deep

20160411 FB 5.7.44 - development only

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.43]

FB 5.7.44 requires OS X 10.6 or higher

(1) build_goodies now contains AppThings.m to support new FBFullStop()

Coding and implementation: Bernie, Brian and Deep

20160408 FB 5.7.43 - development only

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.42]

FB 5.7.43 requires OS X 10.6 or higher

(1) Project Template (i.e. the default project when File>>>New Project is selected) now uses Core Foundation strings.

(2) DEFINEDINCOCOA constant now recognized

(3) Util_UI.incl header updated

(4) Printing folder added to FB Examples. Contains "Print info using NSPrintInfo" and other Cocoa and Carbon (i.e. "PM") printing examples.

(5) Bug fix: FBtoC crashed in fn SendHandleFragmentToOutputStream when an 'end if' was used to terminate an fn and a local variable wasn't dimmed.

(6) Getter/Setter for gCurrFileName and other related code updated to use CFString

(7) ChkBounds() updated to use CFString

Coding and implementation: Bernie, Brian and Deep

20160229 FB 5.7.42

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.41]

FB 5.7.42 requires OS X 10.6 or higher

Thanks to Bernie's prolific coding efforts, we now have new title syntax for some primary FB Statements.

(1) Completely rewritten version of 'Search Apple Headers' application

(2) Similar to the Menu statement changes of FB 5.7.39, the following FB statements now accept either Core Foundation(CF) or Pascal title strings:

```
appearance button
appearance window
button
window
apple menu ( also accepts CFArray of titles )
Edit Field ( FB Help does not reflect this capability yet )
```

(3) The following helper functions are available to get/set CFString text in the menu, appearance window and appearance button statements:

```
For Menus:   MenuSetTitle
             fn MenuCopyTitle
             MenuItemSetText
             fn MenuItemCopyText
```

```
For Windows: WindowSetTitle
             fn WindowCopyTitle
```

```
For Buttons: ButtonSetText
             fn ButtonCopyText
```

(4) FB Help now reflects #2's syntax

(5) CFMenuWndBtnTitles.bas demonstrates #2's CFString capabilities and may be found in FB_5_7_42_Examples/Dialogs and Windows/

(6) Deprecated usage (see N.B. next) based on #2 fixed in FB Editor code, the header FBtoCConsole.incl and (hopefully) the all the example code in FB_5_7_42_Examples

Important: Please Note (N.B.)

FB's next release will not support pascal strings in appearance window, appearance button, apple menu and menu statements, so Fbers are strongly encouraged to upgrade those statements to the new CFString syntax.

Apple has not deprecated Pascal strings(but are discouraged due to limitations) but FB's Runtime Pascal support uses older 32-bit code. The goal is to nudge our code toward 64-bit compatibility as much as possible without disrupting Fbers; small incremental runtime changes should make this easier on everyone.

As a reminder, FB 5.7.42 displays a deprecation message during a build/run: "Warning: Pascal string title is deprecated; use Core Foundation strings instead in line 97 of CFMenuWndBtnTitles.bas"

Coding and implementation: Bernie, Brian and Deep

20160214 FB 5.7.41

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.40]

FB 5.7.41 requires OS X 10.6 or higher

- Fixed: framework included from /Library/Frameworks not working

- Header, Util_UI.incl, added to include Cocoa nib loading & corresponding demo, CocoaNibDemo, added to FB Examples/Cocoa

- FB_5_7_41_Examples/CoreFoundation now includes a demo of replacement CoreFoundation/Foundation string functions(i.e. CF functions to replace, left\$, right\$, mid\$, instr, space\$, string\$)

- Help menu now includes links to the FBtoC web page and FutureBASIC mailing list on associate.com

- Better help message for first-time users seeing the "Could not get read/write access to preferences" message.

201600121 FB 5.7.40

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.39]

FB 5.7.40 requires OS X 10.6 or higher

- Fixed: FBtoC crashed when launched in OS X 10.7

- Added to Tlxb StdCLib.incl: toolbox fn memcpy(ptr dest, ptr src, UInt32 n) = ptr

20160108 FB 5.7.39

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.38]

FB 5.7.39 requires OS X 10.6 or higher

(1) Bernie implemented new CFString/CFArray support for the Menu statement. There is a nice demo in FB Examples/Menus. Thanks Bernie. (now get back to work on that new editor! ;-)

(2) help folder message eliminated for now

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.14]

FB 5.7.38 requires OS X 10.6 or higher

Introduction: The following notes don't adequately capture the significant depth and breadth of FB's (mostly all in FBtoC) changes but hopefully conveys a little of the effort involved.

- With this release, approximately 50%+ of FBtoC is 64-bit (compare to approximately 10% for 5.7.14)
- The editor returns with only minor updates to maintain FBtoC compatibility (Style files, .rsrc resources).
- Lexical code is the next area for FBtoC 64-bitness.
- Several FBers are working on a new 64-bit editor and results are encouraging. Please wish them luck & give thanks; it's a lot of work.

SUMMARY

Items of interest for FBers:

- FBHelp's Begin/end globals now describes the possibility to create C static local variables in local functions.
- Milestone: FBtoC now uses CFString file paths and urls exclusively; there are no FSSpecs.
- FB's generated apps now includes the NSHighResolutionCapable key set to true in their info.plist
- Mid\$ function updated to honor numChars variable. If numChars is zero, Mid\$ returns zero characters and not the whole string as it did before.
- RUN command updated to accept:
 - (a) a CFString literal
 - (b) a CFString variable
 - (c) CFURLRef variable
 - (d) FB Help for Run command updated to reflect new syntax shown
- FB Help Appendix A: File object specifiers updated to recommend CFURLRefs and not FSRefs or FSSpecs.
- 'kill resources' statement isn't supported and is ignored.
- FBHelp's Files\$ doc corrected for options (2) Selecting a Folder and (3) For Selecting a File Name and Folder where a file may be Saved. this was noted by Bernie and RC in a list discussion 18-June-2013.
- Replaced use of old '.rsrc' resources in editor because FBtoC doesn't process .rsrc files
- fn CompilerPath, fn CopyUnixCommandFirstLineResponse and some others changed to accept CFString input
- Support for .rsrc old style resource fork files removed.
 - FBtoC used to combine data from all resource fork files and create an AppName.rsrc with only a data fork. This is now gone
 - Removed copying of 'Localized.rsrc' as it is no longer needed.
 - '.rsrc' resource processing fns removed

Major Areas of Code Conversion to 64-bit

- All file I/O related functions are now use Core Foundation or Foundation. This includes all cached file paths, constants and file reads.
- FB Dynamic arrays replaced with CFMutableArrays
- All "Output Stream" which is all code to copy resources, include files, build files, manage PCH and more
- All error handling
- All TranslateBuild Utilities, including all calls to unix utilities and building buffers for those utilities (i.e. AppendToBuf())
- All FSSpecs removed from all build-related code. Now all CFURLs and CFString file paths.

List of all (mostly) Enhancements:

- Mostly INTERNAL UPDATES ONLY(blue-colored text indicate items of general interest - reverse chronological order with most recent changes first)
 - FBHelp's Begin/end globals now describes the possibility to create C static local variables in local functions.
 - FB dynamic array sFBRuntimeFunctionsToStrip in Symbol Table Implementation replaced.
 - Dynamic array, sIncludedFileNames in Prelexical State replaced
 - More FBtoC internal code updates
 - [01] FB dynamic arrays gCRuntimeIncludeFiles, gFrameworks, gFrameworkPlusHeaders and associated processing in "Conditional Inclusion" replaced with CFMutableArrays
 - [02] Replaced FB string constants in Conditional Inclusion file (and code impacted by them) with CF versions
 - [03] fn WriteTranslatedCodeToFile now accepts CFString input
 - [04] ReadSourceFile now accepts CFString input
 - [05] fn CopyProjectFileURL replaces fn GetProjectSpec
 - [06] Milestone: FBtoC now uses CFString file paths and urls exclusively; there are no FSSpecs.
 - [07] Remaining FB dynamic arrays in Write Translation file converted.
 - [08] FB's generated apps now includes the NSHighResolutionCapable key set to true in their info.plist
 - [09] fn RenameOutput and associated arrays converted
 - [10] Cleanup of fn SendMsgToEditor - seven lines of code reduced to one using Foundation instead of CF
 - [11] fn EmitData converted - this handles FB's DATA statements and associated READ and RESTORE support
 - [12] fn WriteTranslatedCodeToFile - more work
 - [13] fn SaveSourceUserHeaderSearchPaths replaces fn SaveFBSourceFolderReference and uses a CF mutable array instead of an array of FSSpecs
 - [14] fn SetUpFilePathsForThisProject now uses CFStrings.
 - [15] fn WriteTranslatedCodeToFile - updated to use CFStrings and MoveCFStringToRouteBuffer() instead of FB's PRINT
 - [16] ReadSourceFile processing converted but still accepts Str255 input
 - [17] WriteCTypeTableFile converted
 - [18] WriteHandleToFile.(now named: SplitHandleToHeader_C_Line) converted
 - [19] fn SaveSourceUserHeaderSearchPaths replaces fn SaveFBSourceFolderReference. It saves CFString search paths in a global array sCFSearchPaths
 - [20] Mid\$ function updated to honor numChars variable. If numChars is zero, Mid\$ returns zero characters and not the whole string as it did before.
 - [21] fn LastFileExists/fn BuildTempExists now accept a CFURLRef. Calls using it modified to send CFURLRef.
 - [22] fn ProcessOpenedFile now accepts CFStrings and passes them to a primary function: fn TranslateAndBuild
 - [23] FBtoC's Main changed: dictionary now created & sent when this code calls ProcessNotification()
 - [24] fn WriteErrorsFileForEditor now accepts CFURLRef input
 - [25] fn PrintErrorMessageToBuffer updated to use CFStrings and new fn MoveCFStringToRouteBuffer
 - [26] fn WriteErrorsFileForEditor update internally but still receives FSSpec input :-)
 - [27] All PRINT_toBuffer in fn PrintCompilerMessageToBuffer now uses a CFMutableString.
 - [28] The CFString in #27 is moved to the appropriate gFBBuffer via fn MoveCFStringToRouteBuffer which calls FB runtime FBCheckBuffer().
 - [29] Note: this approach relies on the existing ROUTE_toBuffer +(n)/ ROUTE_toScreen logic and doesn't require the new code to figure out which buffer is the current target of the ROUTE. Also note: Since buffer logic is mostly building and expanding a handle to allocated storage it doesn't impede 64-bit compilers.
 - [30] For later: update FBCheckBuffer() and associated code to use allocated pointers instead of allocated handles to avoid all the older SetHandlexxxx calls.
 - [31] Fixed crash on a DisposePtr() in fn ReleaseSourceFileText caused by bug in fn TranslateFileRecursive
 - [32] More work on CompileFiles, CompileSource
 - [33] More work to replace gUnixPathToSourceFolder(pascal) with gCFUnixPathToSourceFolder(CF)
 - Right now fn SetUpFilePathsForThisProject uses CF internally but accepts FSSpecs and Str255. Next step is to have it accept CF input.
 - [34] fn CompileFiles converted to CFString usage and input
 - [35] fn CompileSource converted to CFString usage and input
 - [36] RUN command updated to accept:
 - (a) a CFString literal
 - (b) a CFString variable
 - (c) CFURLRef variable
 - [37] FB Help for Run command updated to reflect new syntax shown in #36
 - [38] FB Help Appendix A: File object specifiers updated to recommend CFURLRefs and not FSRefs or FSSpecs.
 - [39] fn MakeDSym now accepts CFStrings
 - [40] Time display and calcs in fn TranslateAndBuild now CF instead of pascal/carbon
 - [41] fn CopyHelpBookFolderName now accepts, uses and returns CF objects
 - [42] 'kill resources' statement isn't supported and is ignored.
 - [43] fn AddConstantsFromResourcesToSymbolTable is ALMOST 64-bit clean but still passes a pascal string to AddNumericConstToSymbolTable()
 - [44] cleanup from 5.7.21. More direct use of fn CFStringFromPStr(pascalString) instead of letting returned value go stale before being used.
 - [45] from 5.7.16 - "all 'shutdown' and 'stop' code converted to call to 'ShowAlert' followed by [NSApp terminate:nil]"-- cleaned up.
 - [46] fn CopyUNIXPath activated as replacement for older FSSpec-based fn PathForUNIX.
 - [47] fn CopyObject now both accepts CFString input and processes CFStrings
 - [48] fn MakeDir now both accepts CFString input and processes CFStrings
 - [49] fn FileObjectExists now accepts CFStringRef input
 - [50] fn FileObjectExists reverted back to 'test -e' command because it is less code. Internally uses CFStrings.
 - [51] Editor's fn ReadTextFile and fn SaveTextFile updated to use new Cocoa ReadFile() and WriteFile(). Still returns a Handle and accepts an FSSpec.
 - [52] FBHelp's Files\$ doc corrected for options (2) Selecting a Folder and (3) For Selecting a File Name and Folder where a file may be Saved. this was noted by Bernie and RC in a list discussion 18-June-2013.
 - [53] Replaced use of old '.rsrc' resources in editor because FBtoC doesn't process .rsrc files
 - Editor window icons changed from cicon resources to pngs.
 - Menu resources changed to toolbox menu calls (FB menu statements not used but possible for those converting from menu resource use)
 - CFSwapInt16HostToBig() & CFSwapInt16BigToHost() was needed in 2008 but now removed for writing Editor Style Files.incl
 - CFSwapInt16BigToHost() for reading existing version 2 style files remains.
 - FB's Style file I/O using FB's OPEN, READ, WRITE, WRITE FILE, READ FILE changed to Foundation calls
 - Minor update to SystemDirectoryCopyURL() to use signed 'domain' variable and quiet clang warning.
 - fn CompilerPath, fn CopyUnixCommandFirstLineResponse and some others changed to accept CFString input
 - [54] fn CopyResources now accepts CFURL and CFStringRef input.

- Support for copying .rsrc old style resource fork files removed.
 - FBtoC used to combine data from all resource fork files and create an AppName.rsrc with only a data fork. This is now gone
 - Removed copying of 'Localized.rsrc' as it is no longer needed.
 - 'rsrc' resource processing fns removed
 - Some other misc small functions updated and added
- [55] Editor incorrectly labeled text include files (with extension '.incl') as 'resources' inside data.fbproj (within projectname.fbproj)
- Happens only if the '.incl' file does not have a file type of 'TEXT'
 - Building a project once cleans up the projectname.fbproj
 - Editor code that decides whether to write 'include' vs. 'resources' in data.fbproj is fn WriteFileListToProjectDataFork(

[56] All 'shutdown' and 'stop' code converted to call to 'ShowAlert' followed by [NSApp terminate:nil];
 [57] fn SendErrorMessageToLogAndEditor, fn SendInformativeMessageToLogAndEditor converted to CFStrings

[58] FBtoC's 'AppendToBuf()' upgraded to use CFString input. Used this approach (versus just loading a CFMutableString and passing it around) because all the AppendToBuf() code just uses pointers and doesn't invoke Carbon code. This way the logic is maintained and code can continue to call all the command line utilities(especially the compiler, linker, dsymutil, sed and strip). Other commands like 'rm' and 'mkdir' can be switched out for Cocoa but there is no pressing need for that. Tested with FBtoC/Editor source. Bernie's tests on some large projects also works fine.

[59] Lots of minor stuff just to clean up code, update comments and fix broken code.

Coding and implementation: Brian & Bernie

 20150801 FB 5.7.14

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.13]

FB 5.7.14 requires OS X 10.6 or higher.

FB5 bug fixes:

- 'Appearance Window' inappropriately overwrote window attributes when the statement omitted window attributes.
- Editor and Project Manager window resizing was broken in OS X 10.6
- The hidden grow box change, implemented in 5.7.13, now hides grow boxes only for compositing windows.
- Turned off some debugs left on in prior release.

Enhancements:

- OSMajorMinor() returns both major and minor OS version in one call. Example: running in 10.7.5 it returns: major = 10 and minor = 7. Look for a usage example in a later list post.

- Tlxb MacWindows.incl header updated with drawer window and other helpful window calls:

```
// Drawer Window calls now recognized automatically in FB 5.7.14
toolbox fn GetDrawerPreferredEdge( WindowRef inDrawerWindow ) = OptionBits
toolbox fn SetDrawerPreferredEdge( WindowRef inDrawerWindow, OptionBits inEdge ) = OSStatus
toolbox fn GetDrawerCurrentEdge( WindowRef inDrawerWindow ) = OptionBits
toolbox fn GetDrawerState( WindowRef inDrawerWindow ) = WindowDrawerState
toolbox fn GetDrawerParent( WindowRef inDrawerWindow ) = WindowRef
toolbox fn SetDrawerParent( WindowRef inDrawerWindow, WindowRef inParent ) = OSStatus
toolbox fn SetDrawerOffsets( WindowRef inDrawerWindow, CGFloat inLeadingOffset, CGFloat inTrailingOffset ) = OSStatus
toolbox fn GetDrawerOffsets( WindowRef inDrawerWindow, CGFloat * outLeadingOffset, CGFloat * outTrailingOffset ) = OSStatus
toolbox fn ToggleDrawer( WindowRef inDrawerWindow ) = OSStatus
toolbox fn OpenDrawer( WindowRef inDrawerWindow, OptionBits inEdge, Boolean inAsync ) = OSStatus
toolbox fn CloseDrawer( WindowRef inDrawerWindow, Boolean inAsync ) = OSStatus
```

```
// Helpful window-related calls now recognized automatically in FB 5.7.14
toolbox fn IsWindowActive( WindowRef inWindow ) = Boolean
toolbox fn ActivateWindow( WindowRef inWindow, Boolean inActivate ) = OSStatus
toolbox fn CopyWindowTitleAsCFString( WindowRef inWindow, CFStringRef * outString ) = OSStatus
toolbox fn HIWindowGetBounds( WindowRef inWindow, WindowRegionCode inRegion, HICoordinateSpace inSpace, HIRect * outBounds ) = OSStatus
toolbox fn HIWindowSetBounds( WindowRef inWindow, WindowRegionCode inRegion, HICoordinateSpace inSpace, const HIRect * inBounds ) = OSStatus
toolbox fn SetWindowResizeLimits( WindowRef inWindow, const HIRect * inMinLimits, const HIRect * inMaxLimits ) = OSStatus
toolbox fn DetachSheetWindow( WindowRef inSheet ) = OSStatus
toolbox fn HIWindowSetToolbarView( WindowRef inWindow, HIViewRef inView ) = OSStatus
toolbox fn MacGetNextWindow( WindowRef window ) = WindowRef
```

Coding and implementation: Bernie, Brian

 20150622 FB 5.7.13

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.12]

FB 5.7.13 requires OS X 10.6 or higher.

FB5 changes:

- 01 System(_sysVers) changed to use modern Foundation calls instead of Gestalt.
 - N.B.: this call returns exactly what Apple's NSProcessInfo's operatingSystemVersionString provides. It concatenates the major, minor and bugfix version numbers
 - and returns that number as a string: for OS X 10.9.4 major = 10, minor = 9, bugfix = 4, so 1094 would be returned as a SInt32. OS X 10.7.2 returns 1072, OS X 10.10.3 returns 10103 etc. Apple omits the bugfix version if it is zero, so 10.9.0 returns 109 and 10.10.0 returns 1010. This could impact code comparisons if the raw returned value is used as is.
 - This change was prompted by Console log error messages like: "5/6/15 12:30:01.938 PM FutureBasic 5[1755]: WARNING: The Gestalt selector gestaltSystemVersion is returning 10.9.3 instead of 10.10.3. Use NSProcessInfo's operatingSystemVersion property to get correct system version number."
- 02 Windows created with FB's 'Appearance Window' statement now omit the dated visible grow box.
- 03 Garbled warnings for unused functions and unused forward declarations fixed
- 04 'dim as ptr pl, 2' caused crash. Fixed
- 05 FB 5.7.13 source released to public.

Coding and implementation: Bernie, Brian, Michele

Notes: Items #3 and #4 are fixed in 5.7.13 re-released on 9-July-2015

 20150219 FB 5.7.12

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.11]

FB 5.7.12 requires OS X 10.6 or higher.

FB5 changes:

- 01 Many FBtoC source changes adopt Core Foundation and do specific replacements such as pascal string==>CFStringRef & FSSpec==>CFURLRefs.
 - in progress & mostly complete - UNIX source file fns are converted and some feeders and subordinates changed too.
 - in progress & mostly complete - ERROR HANDLING source file fns are either converted to Core Foundation or prepared for it
 - FSSpec replacement with CFURLRef/NSURL started (in progress, incomplete & not activated)
- 02 Fixed: 'dim qs windowref wr' caused a crash.
- 03 Fixed: 'File Not Found' error when a source file directory name contains a higher ASCII symbol (in this case 'f')
- 04 Fixed: Typed pointer treated as untyped generates clang "invalid operands to binary expression" error when used in binary 'and'.

Coding and implementation: Bernie, Brian and Michele

Notes:

 20141117 FB 5.7.11

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.10]

FB 5.7.11 requires OS X 10.6 or higher.

FB5 changes:

- 01 FBtoC's log window rewritten in Cocoa nib and Objective-C
 - (a) Find bar added for searching the log after the build
- 02 FBtoC's menus rewritten in Cocoa nib and Objective-C
- 03 FB5's command menu now has "Analyze" and "Build" options.

- (a) Analyze allows for easy one-off clang analyzes regardless of the Build Settings.
- (b) Build omits the target app launch after the compile/link
- (c) Superfluous "analyze" and "Launch after build" build settings options removed
- 04 Tool tips added to the Build Settings window
- 05 Added: Reminder warning "Compile as 'Objective-C'" is superfluous
- 06 User is warned of a missing Quickdraw framework when choosing Build Settings' base SDK.
- 07 Opening an old project is now smarter and checks if a previous path to a developer folder still exists
- 08 Superfluous FBtoC option "Debug Last App in Xcode" removed.
- 09 More general FBtoC code cleanup
- 10 fn FD_SpecialDirectoryCreateCFURL now allows the programmer to check a returned OSStatus for an error
- 11 Messaging from FBtoC to editor updated to use CFNotificationCenter instead of Apple Events
- 12 NSLog updates:
 - (a) "Compile as Objective-C" removed from NSLog.incl
 - (b) NSLog window bounds now stored in user app's preferences instead of com.berniewylde.nslog.plist
 - (c) Requires 10.6+
- 13 Fix: Path to Developer folder popup failed when more than one path present.
- 14 Fix: Log options now appear only in FBtoC build settings window similar to version 5.7.8
- 15 Fix: 'compilerVersion' now available for use again
- 16 Fix: "Make Xcode without project Stripping" was unresponsive when selected immediately after an "analyze" build of the same code.
- 17 Fix: SystemDirectoryCopyURL(), which supports the File Directory include, updated to support kPreferencesFolderType, kTemporaryFolderType plus a couple bug fixes.
- 18 Fix: Editor incorrectly displays code block (if/else/endif, while/wend local fn/end fn) mismatching error in specific scenario(*). Visual problem only. It does compile.
- 19 Fix: Find Again (CMD-G), after closing source window, find window and project window, opens unexpected "untitled" window and does a find. Should find nothing.
- 20 Fix: Build Settings window fails to display in OS X 10.6.x
- 21 Fix: More helpful editor error message when is unable to open project file. Thanks Michele.
- 22 Workaround: when project with valid developer path is moved to another Mac, build crashes due to invalid path. A warning dialog is now displayed.

Coding and implementation: Bernie and Brian

Notes:

* A line of code exactly 35 characters long triggers a bug when code looks for '#' as the previous character. Turns out 35 is ASCII for the '#' char and comparison with the length byte of a pascal string causes the problem. Code changes for 'if' and 'else'

 20141006 FB 5.7.10

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.9]

FB 5.7.10 requires OS X 10.6 or higher.

FB5 changes:

1. Cleanup from new Build Settings window implementation.
2. Cosmetic bug: log would show "FBtoC: build settings from project" when it was from FBtoC preferences. Right preferences being used but the log message was wrong.
3. New triple slash comment '///' allows FBER to select which comments to send to C/Objective-C generated code
4. FBtoC now generates NSBeep() for 64-bit builds
5. Bug fix: Any included resource with a plist file extension is now correctly put in /Contents/Resources and not /Contents *
6. Bug fix: Crash when Xcode and/or appropriate directories aren't available and Build Settings not consulted by user.
7. Added: Reminder warning "Compile as 'Objective-C'" is superfluous

Comments:

* the info.plist is correctly placed in /Contents

Coding and implementation by Bernie, Brian and Robert C.

 20141003 FB 5.7.9

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.8]

FB 5.7.9 requires OS X 10.6 or higher. Please note the following changes before implementing 5.7.9.

FB5 changes:

1. The Build Settings window has been redesigned and re-written in Objective-C/Cocoa with a Cocoa nib with more options and flexibility.
 - (a) clang is the compiler. Compiler selection is gone.
 - (b) Formerly deprecated 'Allow dim a%, a&, a#, a\$' is now gone. Attempts to use will generate a "Redefined variable".
 - (c) The Settings window does all SDK confirmation/coordination/existence checking. This eliminates build-time checking and also gives the programmer immediate SDK availability feedback.
 - (d) Objective-C compile always used. No other option.
 - (e) Intel architecture always used. PowerPC and Universal builds are gone and not supported
 - (f) First build with PCH does not use anything in "Compiler Options". Subsequent builds use the PCH and any compiler options.
 - (g) Clang analyze works. A bug reported on the list could not be reproduced but current clang analysis output looks normal.
 - (h) Bug fix: "Min Deployment" all OS X releases in popup are selectable. Note: The SDK popup is still limited to those SDKs installed on the machine.
 - (i) "-F/Library/Frameworks" now included automatically in compiler search paths *
 - (j) Table views for 'Path to Developer folder' and 'Compiler options' enable the developer to save and quickly switch to other options without retyping.
 - (k) New FB projects inherit current FBtoC Build Settings
 - (l) Many FB Header, Editor and FBtoC source files were updated to squelch clang warnings about "illegal characters..." All FB source compiles with no warnings.

Comments:

This is the first major FBtoC change in years. Bernie and I are excited the new 64-bit Build Settings coexists nicely with the older Carbon code.

FWIW: Overall code size decreased compared to the prior version.

Coding and implementation by Brian and Bernie.

* For some unknown reason, Xcode 5.1's clang sometimes doesn't find the framework when Build Settings specifies a 10.6 SDK. SDK 10.9 works ok. Clang from Xcode 6 works in either scenario. Bernie's testing was successful in all scenarios but mine wasn't.

 20140703 FB 5.7.8

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.7]

FB 5.7.8 requires OS X 10.6 or higher.

FB5 changes:

Build Settings:

- (1) For the Min OS deployment and Base SDK popup menus: Enabled menu items correspond to installed SDKs on the machine. Disabled menu items indicate SDKs not installed. For example, if the OS X 10.7 SDK is not installed, the 10.7 menu item will be visible but disabled.
- (2) 'Path to Developer folder' defaults to: /Applications/Xcode.app/Contents/Developer

N.B. In order to use 10.7, 10.8 and 10.9 in these popups, users must make SDK changes to support QuickDraw headers that don't exist in the 10.7+ OS X releases. See the FBtoC webpage section titled "Optional QuickDraw headers support in OS X 10.7+" for more information.

New/Updated Headers/Runtime file(s):

NSLog.incl replaces the former FBlog.incl. See Bernie's list post for more NSLog information

New/Updated examples:

- (1) FB_5_7_8_Examples/Miscellaneous/NSLog 1.2.8 -Updates and replaces the former FBlog

 20140430 FB 5.7.7

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.6]

FB 5.7.7 requires OS X 10.6 or higher.

FB5 changes:

The 'Min OS deployment' and 'Base SDK' popups in Build Settings now support OS X 10.7, 10.8 and 10.9.

N.B. In order to use 10.7, 10.8 and 10.9 in these popups, users must make SDK changes to support QuickDraw headers that don't exist in the higher SDK releases. See the FBtoC webpage section titled "Optional QuickDraw headers support in OS X 10.7+" for more information.

New/Updated Headers/Runtime file(s):
Tlxb Aliases.incl: fn FSResolveAliasFile added
Tlxb HICocoaView.incl added

20121120 FB 5.7.6

<http://www.4toc.com/fb/index.htm> [you will need to reload the page if your browser cached the page mentioning FB 5.7.5]

FB 5.7.6 (like 5.7.5 and several releases before it) requires OS X 10.5 or higher.

FB5 changes:
718 Error Missing commas between function params not flagged
197 Error Malformed date/time in editor build log.

New/Updated Headers/Runtime file(s):
FBLog.incl: C printf commands are redirected to FBLog's window.
Minor FB Help updates for dialog function

New/Updated examples:
(1) FB_5_7_6_Examples/Miscellaneous/FBLog_and NSLog/printf test
Demonstrates the redirect of printf with FBLog.incl

(2)FB_5_7_6_Examples/Cocoa/FBNS
Demonstrates Cocoa calls in FB. Both .incl files and FB project version included.

(3)FB_5_7_6_Examples/Files/Bookmarks
Demonstrates bookmarks. Bookmarks are the successor to Alias Manager

(4)FB_5_7_6_Examples/Text/Scrolling text views/YAST folder
YAST(Yet Another Scrolling Text) uses Apple core code to build scrolling text fields.

Notes:
(a) Note the other scrolling text demos in YAST's parent directory
(b) Scrolling text demos in general are, in part, aimed at shortcomings in FB5's Scroll Button
(c) YAST uses Carbon's MLTE which is not recommended by Apple.

(5)FB_5_7_6_Examples/Cocoa/CocoaShell.bas
A basic Cocoa shell(no FB code but can run from FB)
Note: Steve Van Voorst has posted(on fbcocoa but some on the main list) a series of Cocoa/Objective-C demos which are effectively a shell/template. Those exploring Cocoa should check out Steve's demos too.

(6) FB_5_7_6_Examples/Graphics/CoreGraphics (Quartz)/Quartz 1.12

20120918 FB 5.7.5

FB 5.7.5 (like 5.7, 5.7.1, 5.7.2, 5.7.3 and 5.7.4) requires OS X 10.5 or higher.

FB5 changes:
716 Error Util_EUTCKeyFilter.incl misidentified as a resource
717 Error Symbol table overflow. Table size increased by 8k

Please note: FB 5.7.4+ is required for use with OS X 10.8 (see bug #196)

New/Updated Headers/Runtime file(s):
Util_FileDirectoryII.incl:

This new include updates the runtime code for the special directory calls by replacing the now deprecated FSFindFolder() calls. The special directory (e.g. FD_SpecialDirectoryGetXXXXX) parameters have changed slightly. Please see the new example for details. The old include is still available

New/Updated examples:
FB_5_7_5_Examples/Files/File_Directory examples/FileDirectory-SpecDirsUpdated.bas

20120802 FB 5.7.4

FB 5.7.4 (like 5.7, 5.7.1, 5.7.2 and 5.7.3) requires OS X 10.5 or higher.

Please note: FB 5.7.4 is required for use with OS X 10.8 (see bug #196)

FB5 changes:
196 Crash Clicking on project window icons causes crash in OS X 10.8
713 Error Appending pascal string in 'end fn = ' fails
714 Error Error message for non-terminated '#if' is poor

20120528 FB 5.7.3

FB 5.7.3 (like 5.7, 5.7.1 and 5.7.2) requires OS X 10.5 or higher.

FBtoC changes:
704 Error Changes in FBtoC.h not propagated for clang+pch builds
706 Cleanup When save button clicked, fix common user mistakes for 'Path to Developer folder'
708 Error Long variable(symbol) names generate translation and compiler errors
709 Error Make Xcode Project -> Warning: '/' in block comment
710 Cleanup Remove '_A' suffix from translated array names
711 Error Using 'double' variable type with Input fails in FB 5.7.2+
712 Error Cursor not changing from arrow to I-Beam and back in some cases.

New/Updated Headers file:
New - FBLog.incl
Updated - Util_EUTCKeyFilter.incl
Updated - Util_CE.incl

New examples:
FB_5_7_Examples/Miscellaneous/FBLog_and NSLog/NSLog Demo

20120310 FB 5.7.2

FB 5.7.2 (like 5.7 and 5.7.1) requires OS X 10.5 or higher.

Editor changes:
191 Feature Compiler warnings/errors jump to FB code

FBtoC changes:
670 Nicety C files should have UNIX LF line endings only, not mixed with CR
690 Feature Provide FB line info with compiler warnings/errors
691 Feature 'Unused function' warning should jump to definition
692 Error Window title corrupted
695 Feature dialog(-1) returns wndNum for all window events [including _preview]
696 Nicety 'Debug Last App' should attempt to open Xcode 3
701 Cleanup -DDECARBONATE should require -m64
702* Feature Support Xcode 4.3
703 Error Frozen menus from 'window output 0' in DoDialog

Updated Headers file:
Tlxb CFByteOrder.incl

New examples:
FB_5_7_Examples/Graphics/CoreGraphics (Quartz)/CGGradient_demo
FB_5_7_Examples/Graphics/CoreGraphics (Quartz)/RoundRect+CGGradient
FB_5_7_Examples/Graphics/CoreGraphics (Quartz)/UserPaneGrids

* Xcode 4.3 is installed in the Applications folder. To use with FB 5.7.2, Settings > Advanced must indicate the new location.
Path to Developer folder: /Applications/Xcode.app/Contents/Developer
The checkbox 'Use compiler from developer folder' checkbox should be ON.

20120120 FB 5.7.1

FB 5.7.1 (like its predecessor 5.7) requires OS X 10.5 or higher.

Editor changes:

184 Cosmetic Source code drawn in project window
185 Error void functions don't show in function pop-ups
186 Cleanup Bad horiz scrolling of background window by mouse wheel

FBtoC changes:

343 Error gcc error from pointer.offsetConst\$ arg to toolbox Str255 param
505 Error internal error from dim a as ^Point, b&
558 Error Compiler error from block comment in BeginC...EndC
566 Error Comment ending in opt-L eats next line of code
617 Error inc(p) and p++ are different if p is ^Type
619 Error Make Xcode Project strips main() from command-line tool
666 Error 'dim as ..FooType bar' is pointer but should be Handle
667 Error Can't cast address arguments freely and correctly
669 Error Repeated 'clear local' gives internal fatal error
672 Error Build error 'no include path in which to search for Carbon/Carbon.h'
673 Error Default compiler in OS X 10.7 should be clang
674 Error 'void clear local fn Foo as long' should be error
675 Feature Allow spaces in path to Developer folder
676 Error Can't cast to (float)
677 Error Record assignment wrongly disallowed
678 Error Return value from function of type float treated as integer
679 Error 'not(a binop b)' wrongly translated
680 Feature Maximum number of dynamic arrays doubled to 64
681 Error Allow record field as file specifier in 'open' statement
682 Feature Add 'Crash' button to all error alerts
683 Error Buffer overflow when linking huge project
684 Error Generate Localizable Strings fails when compiled as Objective-C
686 Error Unsigned function type wrongly disallowed
687 Error Crash from invalid syntax for 'button', 'appearance button' and 'appearance window'
688 Error FD_SpecialDirectoryXxxx() should create directory if absent

Note:

Projects and standalone builds that used clang with the precompiled header option will encounter an error when first built with FB 5.7.1

Error: clang:
redefinition of 'gFBDynArrayInfo' with a different type [line 115:14 of...]
The simplest workaround is to trash the build_temp folder. An alternative is to switch temporarily to another compiler.

New Examples:

FB_5_7_Examples/Graphics/CoreGraphics (Quartz)/SelfCenteringHIImageView

20111115 FB 5.7

FB 5.7 requires OS X 10.5 or higher.

Editor changes:

140 Error Tab key moves focus to toolbar and prevents subsequent editor window typing
142 Error 'REM' slider in document window's toolbar misbehaves
162 Error Wrong indentation from opt-space between 'end' and 'fn|select|record...'
174 Error Edit menu first item is always 'Can't Undo'
088 Feature Allow mouse wheel/trackpad scrolling of background window
177 Feature Headers ('H') button in doc window's toolbar
179 Feature Adding file to project should make copy if not in project folder
180 Feature Project menu item: New Project File...
181 Feature Help menu item: Search Apple Headers.app

FBtoC changes:

656 Error Hang from missing right parenthesis in 's\$ = edit\$(1'
650 Cleanup Deprecate type-identifier suffices (except '\$') on function names
651 Cleanup Deprecate same name for scalar and array variable: 'dim as long a, a(100)'
652 Cleanup Deprecate vars with same base-name but different type-identifier suffices: 'dim a%, a#, a\$...'
653 Cleanup Deprecate FB 16-bit 'int' type
657 Cleanup Passing pointer instead of record argument should be warning
654 Feature New keyword 'float' as synonym for 'single'
658 Feature Allow pass-by-value records in user-defined functions
665 Feature Allow C's dereferencing syntax for typed pointers
647* Feature New 'if.../else.../endif' syntax
671 Nicety Add Bundle identifier to default Info.plist

New Headers file:

Tlxb CFURLEnumerator.incl

Updated Headers files:

Tlxb CFURL.incl
Tlxb Controls.incl
Tlxb MacWindows.incl
Util_Containers.incl

New functions in Console runtime:

fn ConsolePrintCType(typeRef as CTypeRef) // any CType
fn ConsolePrintCFString(string as CFStringRef) // CFString only; fast

New examples:

FB_5_7_Examples/Files/CFURLEnumerator
FB_5_7_Examples/Files/NSOpenPanel & NSSavePanel
FB_5_7_Examples/Files/Search Apple Headers
FB_5_7_Examples/Graphics/CoreGraphics (Quartz)/ImageEdit skeleton

* The new syntax:

```
if ( 1 == 1 )  
    beep  
else  
    stop  
end if
```

is equivalent to the traditional (and still supported) syntax:

```
long if ( 1 == 1 )  
    beep  
xelse  
    stop  
end if
```

20110912 FB 5.6.2

Major bug-fixes:

641 Error Crash from DynamicNextElement()


```

642 Error      DynamicRemoveItems() attempts to remove non-existent items

Feature introducing type safety to FB:
621 Feature    Allow compiler to warn of some type-mismatched pointer assignments

Cleaning up FB:
636 Cleanup   Deprecate shorthands '& % |' for poke
637 Cleanup   Deprecate 'enterproc ... exitproc'
638 Cleanup   Disallow 'def' and 'usr' function declarations and calls

Improvements in translated C code:
622 Cosmetic  Better translation of 'if ( pointerVar ) ...'
627 Cosmetic  Pointers to scalar types should not be void*
628 Cosmetic  Fewer parentheses in expressions with bitwise operators
634 Cosmetic  Translate '++' as such, not '+= 1'
640 Cosmetic  Translate 'rec = ptrVar' as 'rec = *ptrVar;'

Feature allowing improvement in translated C code:
635 Cosmetic  Void keyword for functions ('void local fn Foo...')

Minor bug-fixes and other changes:
620 Error      Function type mistranslated when ^Type or ^^Type
625 Feature    Allow 'include "BarFolder:foo.c"'
626 Error      Arguments to variadic functions should not be cast
639 Feature    Use of 'fn ToolboxProcedure' as rval should be error
645 Cleanup   Warn against fn CFSTR( stringVar )

```

New example:
FB_5_6_2_Examples/Controls/Custom text browser

20110827 FB 5.6.1

FBtoC general changes:
599 Feature Optional translation of _constants to symbols, not magic numbers
600 Feature New directive: compile [as] "Objective-C"
601 Feature New 'select switch' statement
603 Feature NULL keyword
615* Nicety Warn of long var dereferenced as though pointer
616† Nicety Warn of implied addition of _constants

* Until now, longs and pointers were treated almost interchangeably.
local fn bar(@foo&)
foo.nil& = 0
end fn

This abuse is now deprecated.

⚠ Warning: long var 'foo' used as pointer in line 4 of ...

† The weird syntax '_foobar = _foo_bar' is now deprecated.

⚠ Warning: implied addition of constants; suggested form: '_foo + _bar' in ...

FBtoC changes relating to Make Xcode Project:

```

604 Error      Resource files copied as empty data-fork by Make Xcode Project
607 Feature    Warn of 'include library' framework not passed to Xcode project
609 Error      Xib files wrongly converted to nib by Make Xcode Project
611 Feature    Warn of library linker commands not passed to Xcode project
613 Error      Nibs give ibtool error in Xcode 4 after Make Xcode Project

```

Updated Headers files:

```

Tlbx CarbonEvents.incl [fixes RegisterEventHotKey()]
Tlbx CFCharacterSet.incl [fixes CFCharacterSetXxxx constants]
Others too numerous to list.

```

New examples:

```

FB_5_6_1_Examples/Menu/CFStatusItem
FB_5_6_1_Examples/Miscellaneous/CalCalendar

```

Updated examples:

Too numerous to list.

Deleted example:

```

FB_5_6_x_Examples/Text/SpellCheck [crashes mysteriously]

```

20110722 FB 5.6

Editor feature:

New menu item Tools > Open Headers Folder

FBtoC bug-fixes and features:

```

591 Nicety     Default compiler should be gcc 4.2
592 Nicety     Show compiler path in error message when it cannot be found
593 Cosmetic   Increase ConsoleWindow's height and font size
594 Feature    Complex arithmetic, variables and functions
595 Error      Wrong formatting of signed zero by 'using'
597 Cosmetic   Change 'Max OS features' to 'Base SDK', matching Xcode
598 Error      Crash after multiple 'input' in ConsoleWindow

```

New Headers files:

```

Tlbx CFCalendar.incl
Tlbx complex.incl
Tlbx MDItem.incl

```

Updated Headers files:

```

Tlbx CoreGraphics.incl now has CGDisplayFade functions
Util_CFPrefsCFStrings.incl now has CFPrefsGet/SetCGRect()

```

New examples:

```

FB_5_6_Examples/Calculations/complex arithmetic
FB_5_6_Examples/Cocoa/CocoaControls_nONIB
FB_5_6_Examples/Files/Metadata demo
FB_5_6_Examples/Miscellaneous/CFCalendar demo
FB_5_6_Examples/Miscellaneous/Fade screen to black
FB_5_6_Examples/Text/Print Columnar Data

```

Updated example:

```

FB_5_6_Examples/CoreFoundation/CFArray of Dicts Proj
FB_5_6_Examples/Miscellaneous/FBLog

```

Editor feature:

New menu item Tools > Open Headers Folder

FBtoC bug-fixes and features:

```

591 Nicety     Default compiler should be gcc 4.2
592 Nicety     Show compiler path in error message when it cannot be found
593 Cosmetic   Increase ConsoleWindow's height and font size
594 Feature    Complex arithmetic, variables and functions
595 Error      Wrong formatting of signed zero by 'using'
597 Cosmetic   Change 'Max OS features' to 'Base SDK', matching Xcode
598 Error      Crash after multiple 'input' in ConsoleWindow

```

New Headers files:

```

Tlbx CFCalendar.incl
Tlbx complex.incl
Tlbx MDItem.incl

```

Updated Headers files:
Tlhx CoreGraphics.incl now has CGDisplayFade functions
Util_CFPrefsCFStrings.incl now has CFPrefsGet/SetCGRect()

New examples:
FB_5_6_Examples/Calculations/complex arithmetic
FB_5_6_Examples/Cocoa/CocoaControls_noNIB
FB_5_6_Examples/Files/Metadata demo
FB_5_6_Examples/Miscellaneous/CFCalendar demo
FB_5_6_Examples/Miscellaneous/Fade screen to black
FB_5_6_Examples/Text/Print Columnar Data

Updated example:
FB_5_6_Examples/CoreFoundation/CFArray of Dicts Proj
FB_5_6_Examples/Miscellaneous/FBLog

20110421 FB 5.5.1

Editor bug-fix:
167 Error Strange beep on double-clicking symbol preceded by option-space

FBtoC bug-fixes and features:
584 Error Freeze with 'Wrong number of arguments' error
585 Nicety More compact representation of 'data' strings
586 Error Crash on multi-line strings > 255 chars
588 Feature Advanced settings in drawer
588 Feature 'system long' should allow comma-separated variable list

Updated header:
Tlhx CoreGraphics.incl now includes CGImageSource functions

New examples:
FB_5_5_1_Examples/Cocoa/NoNib_FBtoCocoa
FB_5_5_1_Examples/Cocoa/UITableView_FBtoCocoa
FB_5_5_1_Examples/Controls/bmpImage_btn/
FB_5_5_1_Examples/Controls/HICustomView
FB_5_5_1_Examples/Controls/ThumbnailButton/
FB_5_5_1_Examples/Games/Connect4/
FB_5_5_1_Examples/Graphics/CoreGraphics (Quartz)/CGPathContainsPoint
FB_5_5_1_Examples/Graphics/CoreGraphics (Quartz)/Floodfill
FB_5_5_1_Examples/Graphics/CoreGraphics (Quartz)/ImageResizer
FB_5_5_1_Examples/Text/CoreText styled text
FB_5_5_1_Examples/Text/CoreText with CGColor

Updated example:
FB_5_5_1_Examples/Games/Spandrel_1_4/

20110122 FB 5.5

Major changes:
[1] FutureBasic 5.app is now a self-contained IDE; its supporting file objects (FBtoC.app and the folders build_goodies and editor_support) are inside the app bundle.
[2] An Xcode project is created in the project source folder instead of build_temp as previously. This change makes it less liable to inadvertent deletion. The Xcode project is contained in a folder named <project_name>_XcodeProject.
[3] The User Libraries folder has moved to ~/Library/Application Support/FutureBasic/, alongside Scripts and Style Files.

FBtoC bug-fixes and features:
573 Error Hangs when parsing conditional with @nonexistent_var
574 Feature Make Xcode project in source folder, not build_temp
575 Error FlushWindowBuffer doesn't
576 Error apndstr() wrongly byte-swaps the count field
578 Error val() and val&() return wrong value for 16-digit hex strings
579 Error 'read' gets wrong value for 16-digit hex numbers in 'data' statement
581 Error Link error from Xcode project containing Objective-C code
582 Error NavDialog() releases saveFileName and message wrongly

New examples:
FB_5_5_Examples/Cocoa/DotView_FBtoCocoa
FB_5_5_Examples/Cocoa/GaussianBlur_demo
FB_5_5_Examples/Cocoa/SndPlayer_FBtoCocoa
FB_5_5_Examples/Controls/HISegmentedView
FB_5_5_Examples/Graphics/CoreGraphics (Quartz)/HIImageView_swipe
FB_5_5_Examples/Text/Programmatic HITextView

Updated example:
FB_5_4_9_Examples/Games/Spandrel_1_3

20101116 FB 5.4.8 and FBtoC 1.4.8

Editor bug-fix:
A long-standing 'dangling handle' bug has been found, that could cause a variety of crashes.

FBtoC bug fixes and new features:
570 Error Resources not passed to Xcode project
571 Error Make Xcode Project gives error 'FBtoC_Placeholder_RETURN' undeclared
572 Cleanup Code inflation by numerous gosub/returns*

New example:
FB_5_4_8_Examples/Controls/BrowserCustomItem

*The new implementation of gosub/return does not support optimized compilation. If your code uses gosub/return and needs optimization, you will have to replace every subroutine by an ordinary local fn.

20101008 FB 5.4.7 and FBtoC 1.4.7

FBtoC bug fixes and new features:
536 Cleanup Redundant files copied to XcodeProject
556 Cleanup Editor Errors window garbles multiple errors/comments on same line
557 Error gosub/return broken
559 Error route_toBuffer does not work with ConsoleWindow
560 Feature Support llvm-gcc 4.2 compiler
561 Error _controlkey constant missing
563 Feature Preference setting to control source-file combining

New Headers file:
Util_NibCFStrings.incl

New example:
FB_5_4_7_Examples/Sound/Record_sound_with_QuickTime

Updated examples:
FB_5_4_7_Examples/Games/Spandrel_1_2 (chess)
FB_5_4_7_Examples/Sound/Play Sound file

20100707 FB 5.4.6 and FBtoC 1.4.6

Editor changes:

Two new features mean that you will no longer lose your scripts and style files on updating to new versions of FutureBasic 5.

- [1] Scripts now reside in ~/Library/Application Support/FutureBasic/Scripts. When the editor starts it looks for this folder. If the folder does not exist, it is created and the sample scripts are copied from the current editor_support folder.
- [2] Style files now reside in ~/Library/Application Support/FutureBasic/Styles. When the editor starts it looks for this folder. If the folder does not exist, it is created and the style files are copied from the current editor_support folder.

FutureBasic 5 Help:

This now includes most of the material formerly contained in the FBtoC Help Book. The latter is largely obsolete.

FBtoC bug fixes and new features:

400	Error	preprocessor bug for comments with quote char
463	Feature	pass comments on to C source
519	Error	Misleading error for missing project
522	Feature	Strip unused FB runtime and C runtime functions when making Xcode project
527	Nicety	Cleaner translation of simple 'if' conditions
535	Nicety	Make Xcode Project should set optimization -O0
538	Feature	Send runtime errors to console.log instead of showing fatal alert
539	Cleanup	Compiler warning from A\$ = "(?)"
540	Cleanup	Warnings from Subs Quick Event Loop.Incl
541	Error	write# stringVar gives strange results
542	Error	'line input #' should terminate on <lf> as well as <cr>
543	Feature	Allow casting e.g. x = (double)y
544	Error	instr() treats chr\$(0) and chr\$(1) alike
545	Feature	Allow 64-bit integer literals such as 1ULL, -1LL
548	Error	'Warn of unused functions' sometimes didn't
549	Error	FSCatalogInfo.permission wrongly defined in Headers
550	Error	LP64 types should have size 8 in static expressions
553	Error	inkey\$ for command line tool not implemented
554	Error	'Use precompiled header' breaks with clang 1.5
555	Feature	Allow static string expression (_foo\$ + "bar") as 'output' filename

New Headers file:

Tltx CFNumberFormatter.incl

New examples:

FB_5_4_6_Examples/Controls/CustomShapedControl
FB_5_4_6_Examples/Controls/CustomDrawnControl (Quartz)
FB_5_4_6_Examples/CoreFoundation/PrintCFString
FB_5_4_6_Examples/Files/FileObjectPermissions

Updated examples:

FB_5_4_6_Examples/AppleScript/LaunchMailPressSendII
FB_5_4_6_Examples/Controls/CustomDrawnControl (QuickDraw)
FB_5_4_6_Examples/Files/Droplet apps/Accept folder drop
FB_5_4_6_Examples/Files/Droplet apps/FindInfo
FB_5_4_6_Examples/Files/Droplet apps/FindInfo replacement
FB_5_4_6_Examples/Games/Spandrel 1.1 (chess)

20100406 FB 5.4.5 and FBtoC 1.4.5

Editor changes:

- [1] 'Find in Next File' item (Command-option-G) in Edit menu.
- [2] A crash from typing '#if def _someLongConstant' is fixed.

FBtoC bug fixes and new features:

510	Error	edit\$(without closing parenthesis crashes FBtoC
516	Error	Dim syntax error not flagged
521	Error	Function declared as Str63 is unusable
522	Feature	Strip unused FB runtime and C runtime functions during Make Xcode Project
523	Nicety	Insert spaces around '*' in C translation of typed pointer parameters
524	Nicety	Symbolise boolean constants in C translation
525	Nicety	C translation should have NULL instead of (void*)(0)
526	Cleanup	Remove trigraphs and their support
528	#	Feature Option to build command line tool instead of app
530	#	Feature Allow 64-bit compilation of command line tool
531	Feature	Bounds checking should work for record fields that are arrays
532	Error	Hang when parsing call to undefined function
533	Feature	Allow dereference of pointer that is a record field and a pointer to another record
537	Cleanup	Remove thousands of obsolete predefined constants

see the 'tool' example below. Also see FB5 Help: Appendix J - Command Line Tools.

Function change:

GetFormattedDateTimeAsString() has been replaced by DateFormatCreateStringWithAbsoluteTime().

Note the difference in parameter order:

```
//toolbox fn GetFormattedDateTimeAsString( CFAbsoluteTime at, CFStringRef formatString ) = CFStringRef  
toolbox fn DateFormatCreateStringWithAbsoluteTime( CFStringRef formatString, CFAbsoluteTime at ) = CFStringRef
```

Revised Headers file:

Tltx HIView.incl [fixes definition of HIViewContentInfo]

New examples:

FB_5_4_5_Examples/Command Line Tool/tool
FB_5_4_5_Examples/CoreFoundation/CFString left, right, mid
FB_5_4_5_Examples/CoreFoundation/CFArray of Dicts Proj
FB_5_4_5_Examples/CoreFoundation/CFStrings in FB console
FB_5_4_5_Examples/Dialogs and Windows/ImageView as window background
FB_5_4_5_Examples/Miscellaneous/CFDateFormatter test

Updated examples:

FB_5_4_5_Examples/Miscellaneous/DateExample
FB_5_4_5_Examples/CFPrefs Demos/CF string version/Prefs w/Nib & CE
FB_5_4_5_Examples/CoreFoundation/CFAdditions
FB_5_4_5_Examples/Games/Spandrel 1.0 (chess) [replaces RChess 3]

20100214 FB 5.4.4 and FBtoC 1.4.4

Editor changes:

- [1] Document windows are now live-resizing.
- [2] The text of items in the Errors window can now be copied with Command-C.
- [3] A text selection bug is fixed. It occurred when the mouse was dragged into the left margin. (Thanks to Robert C.)
- [4] Navigation arrows in the Help Book now work as expected.

FBtoC bug fixes:

414	Nicety	'Make Xcode Project' leaves stray .proj in build/NewProjectTemplate/
513	Error	Mouse clicks in compositing window with FB Events and 'on mouse' vector not detected
514	Error	Orphan 'case' statement not identified as error
515	Error	Xcode project template not deleted after Xcode project created
517	Error	mki\$ function can return wrong values
518	Error	edit\$(ef) crashes if the field has not been created
520	Error	Linker warning shows as pseudo error

New date\$ and time\$ functions:

These are now implemented via Core Foundation. They can optionally retrieve both the date and time in one call, and format it in almost any conceivably useful way. See FutureBasic 5 Help for details. The runtime code for date\$/time\$ may be called directly to format any date/time (not just the current one). See DateExample below.

New examples:

FB_5_4_4_Examples/Miscellaneous/DateExample

FB_5_4_4_Examples/Miscellaneous/DockTileBadge
FB_5_4_4_Examples/Files/EnumerateDirectory
FB_5_4_4_Examples/Files/Alias demo
FB_5_4_4_Examples/Text/CFAdditions
FB_5_4_4_Examples/Text/CFAdditions/CFScanner

Updated examples:

FB_5_4_4_Examples/Nib to FB wndNum & btnNum
FB_5_4_4_Examples/App with icon
FB_5_4_4_Examples/Games/RChess_3.0 [replacing RChess_3.0a2]

20091221 FB 5.4.3 and FBtoC 1.4.3

Important Changes for 5.4.3 (note: minor bug fixes and other internal fixes are not shown):

Type Description

New new xib format support added. This is the newer XML nib used in Xcode.
New BeginCCode, BeginCFunction, BeginCDeclaration and EndC are improved methods for including C source directly..
Many examples & documentation updated to use the new BeginCCode BeginCFunction BeginDeclaration keywords.

New .c, .h, .m files are now selectable in "Add to project" file dialog.
New _FBConsoleWndNum constant allows Console window customization .
The user may modify the default window in useful ways, for instance:
include "ConsoleWindow"
window _FBConsoleWndNum, "My tall debug stuff", (5,47)-(500, system(_scrnHeight) - 5)
See new console application demo in Examples/Games

Fix open "A" and open "R" now create the file if it did not exist.
Fix Menu items ending with '!' or '^' or '<' or '/' no longer crash the app.
Fix Linker errors are now passed to editor

New examples:

FB_5_4_3_Examples > CarbonEvents > EventMonitorTarget
FB_5_4_3_Examples > Games > RChess_3.0a2

20091107 FB 5.4.2 and FBtoC 1.4.2

This release is being made early, because of the severity of bug #506 (introduced in 5.4.1).

Editor is unchanged.

Bug-fixes and minor features in FBtoC 1.4.2:

323 Error input "prompt"; x does not work in console mode
422 Error 'dim as long a\$' should be an error
467 Error gCntr = edit\$(1) should work for > 255 chars
471 Error missing '"' gives bad C code
484 Error bad C code from @"\
493 Error str\$("should be translation error")
498 Nicety include "ConsoleWindow" (alternative to include "FBRuntime.incl")
499 Error str\$(fpNum) and 'print fpNum' give different results
500 Error prevent clang Universal build with <10.5 deployment
501 Error in Console Window, 'print fpNum' ignores gFBFloatMaxDigits
502 Error can't access 3D array in record
503 Error prevent array declaration with >4 dimensions
506 Error can't type upper-case letters in edit fields

New examples:

FB_5_4_2_Examples > Calculations > fast sin() and cos()
FB_5_4_2_Examples > Calculations > Expression Evaluator
FB_5_4_2_Examples > Calculations > Measurement of code performance

20091021 FB 5.4.1 and FBtoC 1.4.1

Main changes in FB 5.4.1

[1] Warnings for a common mistake: forgetting to assign a return value (end fn = someValue) to a function whose value is used elsewhere in the program.

```
local fn bar
dim as OSStatus err
err = _paramErr
end fn // oops, should have '= err'
```

```
dim as OSStatus err
err = fn bar
if ( err != _noErr ) then stop "bar() failed" // oops, never get here
```

```
!! Warning: Returned value of bar() is always 0 in line 9 of test.bas
!! Warning: Function should have explicit return value: bar()
```

[2] Choice of compiler. Build settings window has a pop-up menu with four options.
- gcc 4.0 is the default.
- gcc 4.2 is available on 10.5 and 10.6. It cannot build apps to run on 10.3 or 10.4.
- clang is available on 10.6 only. Very fast but somewhat flaky. Best used for development only (change to gcc 4.0 for release build).
- clang (analysis) performs static analysis of your code, giving interesting warnings; nothing is actually compiled.

Minor changes in Editor 5.4.1:

[1] The menu item Project > Remove Item has been deleted.
[2] Bug-fixes in the following areas: text drawing, menus, Find/Replace window.

Bug-fixes and minor features in FBtoC 1.4.1:

476 edit field with filter fn receives Cmd-chars
486 default MinOS deployment 10.4; remove 10.2 target from UI
490 >1-dimension array parameter wrongly translated
492 warn of unused fn prototypes as well as unused fns
494 allow fn instead of def and usr function calls
495 deprecate usr and def function calls
496 terabyte file access; lof(), loc() and rec() return SInt64
497 implement open "N" to give shared access

New examples:

FB_5_4_1_Examples > Graphics > QuickDraw > Image dithering
FB_5_4_1_Examples > Graphics > QuickDraw > Ray Tracing
FB_5_4_1_Examples > Graphics > CoreGraphics > CGGlyphs

20090729 FutureBasic 5.3 and FBtoC 1.3

Issues arising during beta testing of the editor have been resolved. Thanks to all who reported bugs.

Notable changes from FB 5.2:

1. Universal binary editor runs native on Intel.
2. Editor's Build Log shows progress of translation and subsequent build steps.
3. Editor doc windows are useable when toolbar hidden.
4. Editor has Help Book that in part describes FB5, not FB4.
5. New icons for editor, FBtoC, and FB documents (log out/in to see).
6. Find/Change window is resizable.
7. Source code text is drawn with the casing in which it was typed.
8. New projects have extension .fbproj.

Note that FutureBasic 5.3 editor requires FBtoC 1.3 and vice versa; you can't mix and match with earlier versions.

Bug fixes and minor features in FBtoC:

```

421 Error      setting gFBQuit true in DoDialog() doesn't quit
443 Error      unhelpful error msg from malformed _constant
444 Error      string[pointer] should be translation error
445 Error      *.m files not copied during Make Xcode Project
446 Error      string stack bug from 'input z'
447 Feature    'Hide on suspend' preference checkbox
448 Error      Objective-C source files become *.m.m
450 Error      flag missing commas in argument list
452 Error      'Bad case statement' error could force recompilation
453 Feature    allow constants to start with numeral
455 Error      nested index with 'Check array bounds' gives various bugs
456 Feature    change 'Use DWARF' radio button to 'Put line # in app'
457 Error      .c, .m and .h files could open in FB when double-clicked
458 Error      edit$( btnNum ) returns nothing if text len > 255
459 Feature    Save Log As...
460 Nicety     'More warnings' should not include -Wunused-parameter
462 Feature    @"some
string" --> CFSTR( "some
string" )
464 Error      output window wrong for _btnClick in floating window
470 Nicety     explicit error for prototype illegally in local fn

```

New Headers files:
Util_CFUN.incl
Util_Nib.incl

New examples:
FB_5_3_Examples > Carbon in Cocoa
FB_5_3_Examples > Graphics > CoreGraphics (Quartz) > AceOfHearts, ImageMask and ColorMask
FB_5_3_Examples > Files > Accept folder drop

Some old kludgy declarations in Toolbox CFDate.incl have changed to match the official Apple declarations:

```

CFGregorianDateIsValid()
CFGregorianDateGetAbsoluteTime()
CFAbsoluteTimeGetGregorianDate()
CFAbsoluteTimeAddGregorianUnits()
CFAbsoluteTimeGetDifferenceAsGregorianUnits()

```

20090625 Beta version (beta 03) of Universal FB Editor Available

-----Overview-----

The FBtoC team is pleased to announce beta testing for a new universal binary version of the FB Editor. The beta editor is a conversion of the older editor (included with the 'FB_5_2_Package') and thus retains most of the old editor's basic design and functionality. Many of the changes are internal to facilitate a universal binary build but some features were upgraded. Please note: Even though there are many proposed (but not implemented yet) upgrades (many internal) to this new UB editor, the team believes putting a Universal editor in the hands of the FB community is a higher priority than waiting for a 'finished' product. Even though it seems unlikely Apple will remove support of Rosetta in OS X 10.6, they have yet to clarify if it will be included, so having a universal binary editor protects the FB community for now.

-----Changes-----

- (1) Feature: Look up of Apple toolbox calls in Xcode Docs by highlighting the call and using either the contextual menu or the help menu to initiate.
- (2) Bug fix: High level project level disclosure arrow (flipper) showed only "File" label when project closed & re-opened
- (3) Bug fix: Warnings and errors from gcc are mishandled by showDblClickedError() as reported by Andy P. and RC.

-----Installation-----

- (1) Unzip the executable which is named 'FutureBasic_5_3beta' ---do NOT rename
- (2) Put it in the same folder(which is typically 'FB_5_2_Package' for the current release of FB5) as your current 'FutureBasic 5' application.
- (3) Launch 'FutureBasic_5_3beta' instead of 'FutureBasic 5'

Notes:

- (a) The beta editor requires FB5 and does not support FB4 or older FB releases
- (b) The beta editor is built with debugging symbols turned on to provide more debugging information in the event of a failure. The included debugging symbols increase the size of the executable but should not otherwise impact operation.

20090619 Beta version 02 of Universal FB Editor Available

-----Overview-----

The FBtoC team is pleased to announce beta testing for a new universal binary version of the FB Editor. The beta editor is a conversion of the older editor (included with the 'FB_5_2_Package') and thus retains most of the old editor's basic design and functionality. Many of the changes are internal to facilitate a universal binary build but some features were upgraded. Please note: Even though there are many proposed (but not implemented yet) upgrades (many internal) to this new UB editor, the team believes putting a Universal editor in the hands of the FB community is a higher priority than waiting for a 'finished' product. Even though it seems unlikely Apple will remove support of Rosetta in OS X 10.6, they have yet to clarify if it will be included, so having a universal binary editor protects the FB community for now.

-----Fixes for Beta 02-----

- [1] Crash at startup on PowerPC Mac in fn AddToTLBXLList
The crash was the primary reason for releasing another beta so soon after the first beta.
- [2] Compile info window doesn't show *.c files
- [3] Match compile info format from FB 5.2
- [4] Various changes to help menu to reinstate "Apple Web Reference" & other functionality
It is now possible to highlight an Apple toolbox call in the editor window and do a find on Apple's developer site by selecting "Apple Web Reference"
- [5] Wrong fn name at top of editor window
- [6] Initialize fn name and line info fields on doc open
- [7] Re-opening a project should make the PM window front
- [8] Help menu should allow access to miscellaneous files
- [9] horiz scroll bug
- [10] Compile Info window elides some <cr>s

-----Installation-----

- (1) Unzip the executable which is named 'FutureBasic_5_3beta' ---do NOT rename
- (2) Put it in the same folder(which is typically 'FB_5_2_Package' for the current release of FB5) as your current 'FutureBasic 5' application.
- (3) Launch 'FutureBasic_5_3beta' instead of 'FutureBasic 5'

Notes:

- (a) The beta editor requires FB5 and does not support FB4 or older FB releases
- (b) The beta editor is built with debugging symbols turned on to provide more debugging information in the event of a failure. The included debugging symbols increase the size of the executable but should not otherwise impact operation.

-----Reporting Bugs-----

Post to the FB list with the subject: 'FB Editor: yourBug'
To speed the bug reporting process, please include basic environmental information with the bug post. For example, my post would start:
Intel iMac running OS X 10.5.7

20090617 Beta version of Universal FB Editor Available

-----Overview-----

The FBtoC team is pleased to announce beta testing for a new universal binary version of the FB Editor. The beta editor is a conversion of the older editor (included with the 'FB_5_2_Package') and thus retains most of the old editor's basic design and functionality. Many of the changes are internal to facilitate a universal binary build but some features were upgraded. Please note: Even though there are many proposed (but not implemented yet) upgrades (many internal) to this new UB editor, the team believes putting a Universal editor in the hands of the FB community is a higher priority than waiting for a 'finished' product. Even though it seems unlikely Apple will remove support of Rosetta in OS X 10.6, they have yet to clarify if it will be included, so having a universal binary editor protects the FB community for now.

-----Regarding Beta Status-----

Please note: this editor is a ***** BETA *****. This should NOT be interpreted as 'release'. There are identified unfixed bugs and new bugs/failures are possible. Proceed accordingly with saving your work, backups etc.

-----Changes-----

- (1) Mac Intel users will generally notice a lot more zip because of the native executable. Speed increases for opening files (via double clicks on projects/

files and via open dialogs) is particularly noticeable for me.

(2) The Find/Change dialog was improved and now works more reliably. Also, the Find/Change window is resizable.

(3) The split window (index on left) help window is used to display help. The content of help has been revamped to reflect FB5 instead of older FB, but is not complete and a work in progress. Other help manuals used to be available on the help window but are not in this version. Again, in progress.

(4) Several broken features in the old editor were fixed. For example, quadruple click on a long if now correctly highlights the entire structure from long if to end if. This also works for similar Start/End structures like while/wend, local fn/end fn etc.

-----Installation-----

- (1) Unzip the executable which is named 'FutureBasic_5_3beta' ---do NOT rename
- (2) Put it in the same folder(which is typically 'FB_5_2_Package' for the current release of FB5) as your current 'FutureBasic 5' application.
- (3) Launch 'FutureBasic_5_3beta' instead of 'FutureBasic 5'

Notes:

- (a) The beta editor requires FB5 and does not support FB4 or older FB releases
- (b) The beta editor is built with debugging symbols turned on to provide more debugging information in the event of a failure. The included debugging symbols increase the size of the executable but should not otherwise impact operation.

-----20090612 FutureBasic 5.2 and FBtoC 1.2-----

Editor is unchanged.

Bug fixes in FBtoC:

431	Closed	string stack bug from 'open "0", 1, name, @parFSRef'
432	Closed	'include "foo.c"' could give bad C code
433	Closed	log window shrinks each time FBtoC is launched
435	Closed	out-of-order menu items become spacers
436	Closed	hang from 'if 1 then pen 1 else pen 2'
437	Closed	Overlong path to project file should give helpful error
438	Closed	allow 16 parameters in function definitions
439	Closed	array bounds error translating 'for i=1to 6'
441	Closed	spurious translation error from 'stop fSpec.name'
442	Closed	stale gcc warning shown by editor

New feature:

p.0@ dereferences a pointer to a 64-bit integer (like p.0& for 32 bit)

New examples:

FB_5_2_Examples > Graphics > OpenGL > OpenGLTestBed
FB_5_2_Examples > Graphics > QuickDraw > (de)convolve img
FB_5_2_Examples > Miscellaneous > SortDescriptor

-----20090321 FutureBasic 5.1.5 and FBtoC 1.1.10-----

Editor changes:

- [1] An unsaved standalone file can now be built and run (Command-R); see note 1 below.
- [2] Compile Info window shows progress of both translation and compilation.

Bug fixes in FBtoC 1.1.10:

426	Closed	inkey\$ does not return
427	Closed	val&("01") and val&(" 1") should return 1
428	Closed	spurious error from 'if...then edit\$(1) = "" else stop'
429	Closed	'text' statement with no parameters hangs FBtoC
430	Closed	Tlhx glu.incl could not be included

New example:

FB_5_1_5_Examples > Text > HITextView+FontPanel > HITextView
An HITextView with the FontPanel allows multistyled text with shadow, colour, double-underline and other eye-candy.

Note 1:

This is a normal Run in every way, except that the various files required by FBtoC are in your TemporaryItems folder, which (in OS X 10.5) is somewhere like /private/var/folders/88/88AKX+F62RWmd++8ZOxBkU+++TI/TemporaryItems.

-----20090216 FutureBasic 5.1.4 and FBtoC 1.1.9-----

Editor 5.1.4 changes

1. Scripts '#if 1/#endif' and '#if 0/#endif' fixed.

FBtoC 1.1.9 changes

1. Typed functions without a suffix are allowed, for example
local fn Name as Str255
end fn = "Ann"
2. FB4 kludges removed from toolbox declarations (see below).
3. The setting 'Use precompiled header' works for universal builds.
4. The build system determines which pieces of a large program need to be compiled, and compiles only those pieces.

Bug fixes

415	Closed	Make Xcode Project wrongly quotes/escapes nib names
417	Closed	dropping file on FBtoC spoils log window position
420	Closed	Allow files with no OSType to be opened.
424	Closed	input # bug
425	Closed	record # should set file mark when extending

FB4 kludges removed

The Carbon C definition of these five functions includes a 64-bit parameter, which in FB4 had to be split into two 32-bit halves:

HIViewFindByID()
HIViewSetID() // rarely used
CreateMouseTrackingRegion()
HIViewNewTrackingArea()
HIViewChangeFeatures() // rarely used

You will have to change your code from, for example:

```
fn HIViewFindByID( fn HIViewGetRoot( w ), myHIViewID.signature, myHIViewID.id, @myView ) // old syntax
to the simpler correct form:
fn HIViewFindByID( fn HIViewGetRoot( w ), myHIViewID, @myView ) // FB5.1.4 syntax
```

-----20090109 FutureBasic 5.1.3 and FBtoC 1.1.8-----

Editor 5.1.3 is unchanged from 5.1.2.

FBtoC 1.1.8 changes:

'on FinderInfo fn xxxx' implemented [see FB_5_1_3_Examples > Files > FinderInfo]
The Headers file Util_CE.incl has been extensively commented for easier use.

New Headers files and examples:

Util_EUTCKeyFilter.incl	[see FB_5_1_3_Examples > Text > EUTC Key filter]
Util_Toolbar.incl	[see FB_5_1_3_Examples > Carbon Toolbar > The easy way]
Tlhx HITheme.incl	[see FB_5_1_3_Examples > Controls > HITheme_demos]
Tlhx CoreText.incl	[see FB_5_1_3_Examples > Text > CoreText xxxx]

CoreGraphics headers changes

The special FB4 syntax for a handful of CG functions (CGRectMake, CGRectOffset, CGContextGetCTM ...) has been discarded. You will need to change the way you call these functions in your source code. For example:

```
fn CGRectMake( r, 0.0, 0.0, 10.0, 20.0 ) // old FB4 syntax
r = fn CGRectMake( 0.0, 0.0, 10.0, 20.0 ) // new FB5 syntax similar to C
```

Additions to the CG headers:

```
CGDataProvider.incl
CGError.incl
CGFont.incl
CGGradient.incl
CGPath.incl
CGPDFArray.incl
CGPDFDictionary.incl
CGPDFDocument.incl
CGPDFObject.incl
CGPDFPage.incl
CGPDFStream.incl
CGPDFString.incl
```

The FB4 'local fn' wrappers have been removed from all CoreGraphics functions. This change makes 'fn' optional when calling CG procedures (functions that do not return a value).

```
fn CGContextFillRect( ctx, r ) // 'fn' formerly required; now optional
CGContextFillRect( ctx, r ) // new simpler syntax
```

Bug-fixes:

```
370 button statements uncheck a checked checkbox
404 scroll button mishandles Rect param at creation
406 key chars with high bit set are negative in _evKeys dialog event
409 I-beam cursor persists after click in FBtoC Log window
410 gcc error from struct/record field named 'index'
412 gcc error from pi if prefs 'Allow dim a#,a$,a%'
413 val&() should return 0 if first char is alphabetic
```

20081117 FutureBasic 5.1.2 and FBtoC 1.1.6

Important change in HandleEvents

The default behaviour of HandleEvents is now to block (i.e. not return) unless an event is dispatched. It is no longer necessary (but is harmless), to suppress null events with:
poke long event - 8, 0xFFFFFFFF // no null events

The original behaviour of returning 30 times a second, if required for old polling code, can be restored by:

```
poke long event - 8, 2 // null events every 2 ticks, like FB4
do
fn PollForSomething
HandleEvents
until gFBQuit
```

Editor (FutureBasic 5.1.2) changes

Fixes to text display in the Errors window.

A bug has been fixed that caused the cursor to change to an arrow over the Scripts Palette even when the latter was hidden.

FBtoC 1.1.6 changes

1. Default behaviour of HandleEvents changed.
2. The rarely-used 'system' statement is no longer equivalent to 'end'. Please change your code to use 'end'.
3. The 'menu' statement now allows an optional 5th parameter (commandID).
4. New/revised Headers files: Tlxb CFBase.incl, Tlxb CFString.incl, Tlxb CFArray.incl, Tlxb CFData.incl, Tlxb CFDate.incl, Tlxb CFNumber.incl. These header files are more complete than previously; if you get redefined function errors for your own CFxxxx declarations, just comment out your declarations.
5. New Headers file: Util_Containers.incl [for use see FB_5_1_2_Examples > Containers > Util_Containers example].
6. New Headers file: Util_Toolbar.incl [for use see FB_5_1_2_Examples > Carbon Toolbar > Toolbar Example].
7. FB_5_1_2_Examples > Sorting [FB3/FB4 CD examples updated for FB5]
8. build_temp folders created by FBtoC are excluded from Time Machine backups.

Bug fixes

```
390 Closed left$$, mid$$, right$$ give _zTrue when comparing chars
391 Closed mid$$ ( c, 1 ) gives "comma expected error"
392 Closed scrollbar of console app lacks down-arrow
393 Closed garbled error from proc "NoSuchFunction"
394 Closed prHndl.prInfo.rPage fields are byte-reversed on Intel
396 Closed default edit field is _framedNoCR instead of _framed
398 Closed can't access 2-D array inside record
399 Closed def apndstr has endian bug (on Intel)
401 Closed crash from print after front window closed
402 Closed 'Array bounds error' alert shows as 'Quit'
```

20081020 FutureBasic 5.1.1 and FBtoC 1.1.5

Editor (FutureBasic 5.1.1) changes

1. Script Palette replaces Script submenus.
2. Project window has Headers button, which opens the Headers folder in the Finder.

FBtoC 1.1.5 changes

1. New File menu item: Make Xcode Project.
2. New feature and setting: Warn of unused functions.
3. Domain and folder type constants (_kUserDomain, _kApplicationSupportFolderType...) are now defined.
4. New Headers file Util_CE.incl provides a simplified method to implement Carbon Events in your code; for demos see FB_5_1_1_Examples > CarbonEvents.

Bug fixes

```
314 Closed make strings in *.c runtime localizable
378 Closed elided error message from include library "NoSuchFile"
381 Closed editor ignores some translation errors
382 Closed aliased source file prevents translation of main
383 Closed editor's Stop button does not affect build by FBtoC
384 Closed I-beam cursor stays after click in Prefs 'More gcc options'
385 Closed unusable CreateMouseTrackingRegion() in Tlxb CarbonEvents.incl
386 Closed 'picH = picture' gives translation error
387 Closed _useWFont fails for some appearance buttons
```

20090924 FutureBasic 5.1 and FBtoC 1.1.4

Important change in FBtoC

Until now, the build process would automatically copy every nib (*.nib) from the source folder into the built app package. This greedy and surreptitious copy risked the inadvertent inclusion of files like "OldRubbish.nib" and "embarrassingly bad.nib". Now, nibs are ignored unless specified in an 'include resources' statement or in the editor project window (see below).

This means that every project or standalone file that uses a nib interface needs modification. The most straightforward fix is to add a line of code resembling this:
include resources "main.nib"
for each nib used by your project.

Editor (FutureBasic 5.1) changes

1. New menu items give better integration with FBtoC

20080912 FutureBasic 5 and FBtoC 1.1.3

The FutureBasic 5 editor is FBtoC-only as previously discussed.

After downloading the FB 5 package, please double-click FBtoC.app then click Open in the "downloaded from the Internet" warning dialog. Otherwise this dialog may interfere with your first Run from the editor.

20080722 FBtoC 1.1.2

1.1.2 build 140

Two helpful additions:

- [1] VAList - Variable number of function arguments support and examples added
- [2] FBShell - Basic program shell with more modern nib and carbon events

Bug Fixes:

- 362 Closed 140 Error if 1 then does not produce FBtoC error message
- 363 Closed 140 Error menu 127, 1, _disable does not work in FBtoC
- 364 Closed 140 Error Menu cmd keys don't auto repeat when created with FB menu statement
- 366 Closed 140 Error Record# should extend fork size without error

20080618 FBtoC 1.1.1

New Features:

(1) The new NavDialog function, an FBtoC-only option, is similar to the FB Files\$ keyword but provides substantially more functionality. NavDialog offers new functionality via Navigation Services but the details are hidden in the C runtime, so it is easy to call NavDialog (and several other NavDialogxxxxxx helper functions). Several resources are available for learning to use NavDialog(). They are:

- (a) The demos - see Examples/Files/NavDialog_Demos,
- (b) FBtoC help - see Language Enhancements
- (c) Browsing the NavDialogxxxxxx functions in FilesDollarFunction.c and some new FSRef / CFURL helpers in FileHandling.c (both in the build_goodies folder)

(2) FBtoC now uses NavDialog(_kNavDialogChooseFile + _kNavDialogSheet,...) internally.

(3) FBtoC Help for include statement revised

(4) Internal AppleScript arrangements simplified for debugging in Xcode

Bug Fixes:

- 246 Closed PM sheets beep in FBtoC-built app
- 359 Closed Appearance button btnNum, _activeBtn did nothing
- 360 Closed NavDialog sheets retract at once

Closed files\$(_FSRefSave,...) now returns parent folder ref always

- 361 Closed open "O", 1, s, @fsRef fails if fsRef is parent folder but file exists
- 362 Closed Architecture: Current Mac should disable OS min/max pop-ups

20080525 FB 4.4.3

The change most folks will appreciate is the ability to compile directly with either the FBtoC translator/gcc compiler or with the FB compiler. It is available now on the command menu.

Download FB 4.4.3 here

New Features for FutureBASIC 4.4.3 (aka: changes since 4.4.2)

4.4.3 (24 May 2008)

[1] Command menu has Run [FBtoC]

Note: if you're planning to use the new Run [FBtoC] command from within the FB Editor, please refer to the "Connecting to FBtoC" document for one-time setup steps.

[2] Obsolete 680x0 and FAT items removed from command menu

[3] New Headers files, matching FBtoC:

Subs FloatByteSwapping.incl

Tlxb AppleHelp.incl

Util AppleHelp.incl

[4] Rntm Appearance.incl update fixes little-arrows-wrong-way-counting

[5] Subs Appearance.Btns.Incl update supports viewsize for scroll bars.

[6] Subs Compiler.Incl update suppresses double-beep when adding too much text to an edit field.

[7] Subs Graphics.incl fixed for PPC compile.

[8] Definition of CFStringAppendPascalString corrected in Tlxb CFString.incl.

[9] FB Compiler v4.6 no longer emits "Warning: Your source code is ignoring the result of this toolbox function".

Special note for OS X 10.5 users

Before using FB 4.4.3 for the first time, locate the new FB Compiler.app in the Finder (the path is: FB_pkg_4_4_3/FB Extensions/Compiler/FB Compiler.app) and launch it by double-clicking its icon. If an alert appears ("FB Compiler.app is an application which was downloaded from the Internet. Are you sure you want to open it?"), click the Open button.

P.S. Currently, the 4.4.3 version is only available in English.

20080524 FBtoC 1.1

New features:

- [1] files\$ supports _FSRefFolder and _CFURLRefFolder as alternatives to _FSSpecFolder.
See Examples > Files > open/files\$ FSSpec/FSRef/CFURLRef

[2] MakeIntoFBWindow() has been improved to handle tracking controls correctly (scroll bars, little-arrows and sliders, both live and non-live).

The improvements required the code to be moved to the C runtime instead of being user FB code.

See Examples > Nib to FB wndNum & btnNum > Nib demo.bas

See also NibToFB.incl in the same folder, in which MakeIntoFBWindow() is now declared:

```
toolbox MakeIntoFBWindow( WindowRef w, SInt32 wNum )
```

The syntax fn MakeIntoFBWindow(wRef, wNum) is unchanged.

[3] FBtoC can now be invoked from the FutureBASIC editor (v4.4.3 whose release is expected soon).

New Header files:

Tlxb AppleHelp.incl and Util AppleHelp.incl

Bug fixes:

- 355 Closed files\$(_FSSpecFolder,...) should produce no-name variant FSSpec
- 356 Closed nib sliders and little-arrows don't send dialog events
- 357 Closed edit\$(n) = "ABC" blows string stack
- 358 Closed 'input #1, longvar' gives 0 for -ve number strings

This procedure converts a nib-based window and its controls to behave as though created programmatically with 'appearance window' and 'appearance button' statements. After a call to MakeIntoFBWindow(), the window has an FB wNum and its controls have FB btnNums, and generate FB dialog events.

20080508 FBtoC 1.0.9

The 1.0.9 release of FBtoC (English only) repairs some important bugs.

The French localization, already outdated in 1.0.8, became severely dysfunctional in 1.0.9 and has been temporarily removed.

Bug fixes:

- 351 Closed 'input #fileID, stringVar, singleVar, doubleVar' gives null string and 0s
- 352 Closed 'on mouse fn DoMouse' interferes with button handling
- 353 Closed little-arrows and sliders should give correct FB dialog events
- 354 Closed 'edit field' blows string stack

20080424 FBtoC 1.0.8

New special constant _forLoopsAlwaysExecutedAtLeastOnce:

```
for j = 1 to 0
  print "Never get here"
next
override _forLoopsAlwaysExecutedAtLeastOnce = _true
```



```
for j = 1 to 0
  print "Do get here (legacy FB behaviour)"
next
```

Division symbols:

```
/ and double-backslash: always mean the same thing (integer division if appropriate)
\                          always means force floating point division
```

Bug fixes:

```
345 Closed      close box doesn't close window
347 Closed      performance issue with container += string + string
348 Closed      crash from edit$( 1, _maxInt, _maxInt ) = "a"
349 Closed      performance issue with hex$()
```

20080327 FBtoC 1.0.7

New feature:

```
- for/next loops may be executed zero times
for j = 1 to 0
  print "Never get here"
next
```

New examples:

```
Examples > Calculations > Counting for 1 second
Examples > CoreGraphics (Quartz) > CGBitmap demo
```

New documentation:

```
Build system caching.rtfid [a file in the FBtoC_1.0.7 folder]
```

Bug fixes:

```
284 Closed      can't assign a string const to another
327 Closed      'print myUInt64Var/1024/1024/1024' prints 0
336 Closed      line continuation bug
337 Closed      def open "TYPE" does nothing
338 Closed      default file type should be 'TEXT'
339 Closed      crash from 'print test.r.left'
342 Closed      some labels become illegal C code
344 Closed      wrong number of decimal places sometimes printed
```

Open bugs:

```
314 Open        make strings in *.c runtime localizable
323 Open        input "prompt"; x does not work in console mode
343 Open        gcc error from pointer.offsetConst$ arg to toolbox Str255 param
```

20080307 FBtoC 1.0.6

New feature:

```
- menu item "Move build_temp to Trash"
```

New examples:

```
Examples > Mandelbrot set
Examples > Carbon Toolbar > Toolbar.proj
```

Bug fixes:

```
077 Closed fix the 'as pointer/Handle to type' mess
212 Closed crash when global var 'err' is used (OS X 10.4)
216 Closed app gets generic icon if project name contains '&'
329 Closed 'stop' decrements the string stack
330 Closed 'picture' statement can crash in OS X 10.2.8
331 Closed gcc error from global vars j0, j1, jn, y0, y1, yn
332 Closed Template error if FB-built app is in source directory
333 Closed spurious error from 'long color' inside 'long if' block
334 Closed error: 'index' redeclared as different kind of symbol
```

Notes:

With bug #077 closed, you can now write code like this:

```
'-----
#define MyFunkyType as Handle to Rect
```

```
begin record DemoRec
dim as MyFunkyType h1, h2
dim as pointer to Rect p1, p2
end record
```

```
dim as DemoRec myFunkyVar
```

```
myFunkyVar.h1 = fn NewHandle( sizeof( Rect ) )
myFunkyVar.h1..top = 99
'-----
```

which was impossible in FB owing to bugs (that remained unfixed throughout the compiler's entire lifespan).

FBtoC also fixes this related FB bug:

```
'-----
dim myPtrToSpec as .FSSpec
print "Should be 4 because it's pointer:", sizeof( myPtrToSpec )
'-----
```

20080222 FBtoC 1.0.5

New features:

```
- French localization of FBtoC
- simplified debugging with Xcode, via menu item File > Debug Last App
```

New example:

```
Examples > Debug > DemoCrash
```

```
DemoCrash crashes at the line:
poke address, value.
```

```
When run under Xcode's debugger, the crashing code and variable
values are shown
```

20080205 FBtoC 1.0.4

New features:

```
on edit fn MyDoEdit
tekey$ function
support for localization of string literals
```

New example:

```
Localizable strings
```

Bugs fixed:

```
304 Closed      int(1/3*100) evaluates as 0
306 Closed      print #fileID spc( 10 ) prints nothing to file
307 Closed      comma-tabbing and tab() broken for print #fileID
308 Closed      remember Finder-dropped file for 'Open Last File' command
310 Closed      pos( fileID ) returns nonsense
311 Closed      closing a _scrollVert button makes window( _width ) off by 1
313 Closed      close box should not close window if 'on dialog fn
DoDialog' is in effect
318 Closed      text statement can't be followed by a colon
319 Closed      gcc warning from picture statements
```

320 Closed inc() / dec() can't be applied to record fields

20080129 FBtoC 1.0.3

Updated Headers file:
CFIndex.incl

Bugs closed:
279 Closed _scrollVert and _scrollHoriz scroll buttons not fully working
301 Closed custom-sized string vars can wrongly be Str255
302 Closed button value can't be set with 'appearance button...'
303 Closed files\$(_FSSpecFolder...) returns invalid FSSpec

20080126 FBtoC 1.0.2

Updated Headers file:
CFIndex.incl

New feature:
_scrollVert and _scrollHorz scroll buttons can now be created. They resize automatically with the window. But the apparent window size (from window(_width) etc) is not yet adjusted for presence of scroll bars.

Bugs closed:
283 Closed problems from source file with long name
293 Closed allow drag and drop from Finder
296 Closed 'input' gets character on key-up
297 Closed DIM 4 MTBold% should be a translation error
298 Closed _wndRefresh, _wndActivate FB events not received
299 Closed 'unterminated comment' gcc error for some division expressions
300 Closed FB dialog handler signals _evKey instead of _evCmdKey

20080116 FBtoC 1.0.1

Now implemented:
page lprint, read dynamic, write dynamic, scroll

New section in FBtoC Help:
Byte order

Headers files now available:
Subs AppleScript.incl, Tlbx CFByteOrder.incl

Bugs closed:
285 Closed 'window 1' resets print position to top left
287 Closed 'on mouse' disabled
289 Closed printed text wrongly split over several lines
292 Closed built app has old creation date
294 Closed last member of STR# resource not retrieved
295 Closed crash from input "test"; x

20080102 FBtoC 1.0

Changes since preview 1a126:
Translates 'on timer fn DoTimer'
SInt64 and UInt64 are now integer types, not records

Bugs closed:
265 'dim dynamic' fails for custom-sized string array
281 custom-sized string vars can't be suffixed with '\$'
282 allow kEventRawKeyRepeat to reach fn DoDialog as _evKey