



## Xojo Design Mistakes

Bob Keeney  
Founder  
BKeeney Software Inc.

## Alternate Title:

Thankfully time travel hasn't been invented otherwise my future self might come back to murder me because of stupid decisions.



## Coding Mistakes

"The key to efficient development is to 'make interesting new mistakes'." - Tom Love

## Using Goto

```
Run
for i as integer = 0 to lstPetNotes.ListCount-1
    If m_bWeAreClosing Then GoTo bail ←
    Dim oNote As clsPetNote = lstPetNotes.RowTag(i)
    oNote.OrderNumber = i
    Call oNote.save(g_Employee.id)
Next
bail: ←
Exception err
    ExceptionReport(err, CurrentMethodName)
```



## Using DoEvents

- Used to start the event loop
  - For Console Apps only
- DoEvents in Desktop When:
  - Never
  - I repeat never

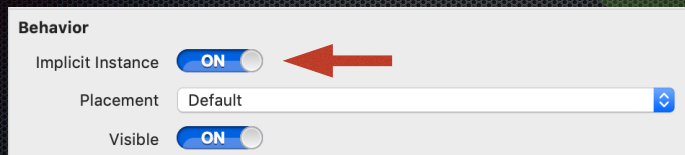
## Using Too Many Variants

```
Dim vValue1 As Variant = "1"
Dim vValue2 As Variant = 2
Dim vValue3 As Variant = 3.0
Dim vTotal1 As Variant
Dim vTotal2 As Variant

vTotal1 = vValue1 + vValue2 + vValue3
//123.0

vTotal2 = vValue3 + vValue2 + vValue1
//5.01
```

## Window.ImplicitInstance = true



## Poor Variable Names & Multiple Declarations on One Line

```
dim c as Color
dim w,h,x,y as Integer
dim p,p2 as Picture
dim obj as Object2D
dim s,r as RGBSurface

Dim w2,h2,w3,h3,wSource,hSource As Integer
Dim a,t,d,tx,ty,i, RadianAngle, sn, csn,tySin,tyCos As Double
dim outX,outY as Integer
dim xCache() as Double
dim xx,yy as Integer
```



## Tight Loops/UI Freeze

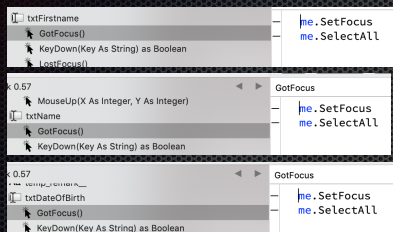
```
Dim iMax As Integer = 10000
ProgressBar1.Maximum = iMax
For i As Integer = 0 To iMax
    //This could take a while
    ProcessLine(i)
    //Update the ProgressBar
    ProgressBar1.value = i
Next
```

## Abusing AddHandler

```
Dim dlg As New dlgTRDepartmentEdit
AddHandler dlg.dismissed, AddressOf DialogDismissed
dlg.display oRecord
```

```
DialogDismissed(dlg as dlgTRDepartmentEdit)
RemoveHandler dlg.dismissed, addressof DialogDismissed
LoadList
```

## Not Subclassing Controls

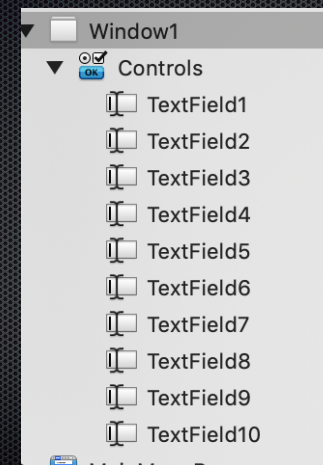


### TextField Subclass

#### Handle Events:

GotFocus  
KeyDown  
LostFocus  
Others....

## Using Default Control Names





## Using Integer Instead of Enum's

```
' Set to left aligned using integer constant.  
Dim iSomeValue As Integer  
iSomeValue = FormattedText.DISPLAY_ALIGN_LEFT  
  
TestDisplay.SetDisplayAlignment(iSomeValue)
```

```
setDisplayAlignment(alignment as integer)
```

```
' Set the display alignment.  
DisplayAlignment = alignment
```

```
' Set to left aligned - Using eNum  
Dim NewValue As FormattedText.Display_Alignment  
NewValue = FormattedText.Display_Alignment.Left  
  
TestDisplay.SetDisplayAlignment(NewValue)
```

## Worthless Comments

```
' Are we in the read only mode?  
if readOnly then return
```

```
' Get the proper state change value.  
value = not selectionBold
```

```
' Is there anything selected?  
if doc.isSelected then
```

```
' Get the selection range.  
doc.getSelectionRange(startPosition, endPosition)
```

```
' Get the original content.  
originalContent = doc.getXML(startPosition, endPosition)
```

## Magic Numbers

```
iXCor = Floor(iCardWidth * .55)  
iYCor = (iCardHeight \ 2) + iTopOffset  
iWrapWidth = Floor(iCardWidth * .45)  
gPrinter.TextSize = 14*(ratio.Right/72)
```

```
dim w as Window = self.Window  
while true  
  x = x + w.Left  
  y = y + w.Top  
  if w isa ContainerControl then  
    w = ContainerControl(w).Window  
  else  
    exit while  
  end if  
wend
```

```
x = x + 10
```

## If Statements that Span the Entire Method

```
m_AutoReminder = new clsAutoReminder  
if m_AutoReminder.Load(iIndex) then  
  //Do a bunch of stuff  
else  
  //Load Failed. Exit...  
  Return  
end if
```

```
m_AutoReminder = new clsAutoReminder  
If m_AutoReminder.Load(iIndex) = False Then  
  //Load Failed. Exit  
  Return  
End  
  
//Do a bunch of stuff
```



## Multiple Things on a Single Line of Code

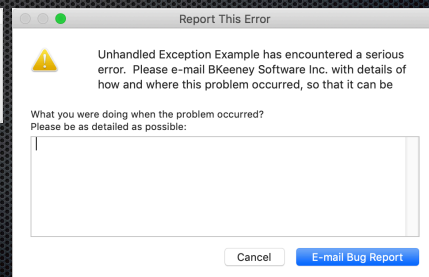
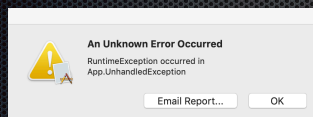
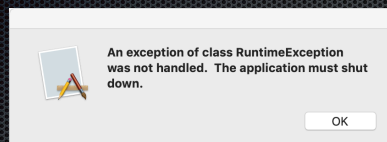
```
If m_bWeAreLocked Then Return True
```

```
SpecialPopupMenu.ListIndex = If(firstIndent < 0, 1, 0)
```

## Using Multiples Arrays Rather than Data Class

```
▼ ✓ Properties  
  ardtmDateOfBirth() As Date  
  1 ariUserID() As Integer  
  "" arsEmail() As String  
  "" arsFirstName() As String  
  "" arsLastName() As String
```

## Not Handling Exceptions



## Not Checking Text Encoding

- Inside computer:
  - Known Encoding
  - Default is UTF8
- Outside computer (DB, Web Service):
  - Unknown Encoding
  - Use DefineEncoding/ChangeEncoding
- Check out Kem Tekinay's session on Encoding!



## Loading Too Much Data into UI Controls

- 10,000 rows in a Desktop listbox is bad
  - Slow/User has hard time navigating
  - Think about "Paging" Data instead
- 10,000 rows in a WebListbox is worse!
  - Awful slow (be better in Web 2.0)/User has hard time navigating
  - Think about "Paging" Data instead
- PopupMenus
  - Keep list to a 'screen'
  - Consider alternate UI (dialog with ListBox w/search)

## Huge Methods Are Bad

- Huge methods (i.e. > 1000 lines) are bad
- Hard on the compiler
- Example:
  - Window.open event had over 10,000 lines of code
  - Dozens of loops and If-Then statements
  - DoEvents and Goto's

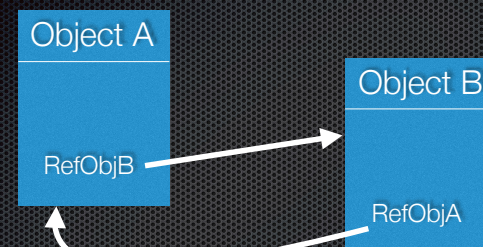
## Too Many Nested Loops and If Statements

```
Run
oEntry.dAmount = -(rs.Field("amount").DoubleValue)
oEntry.sPaymentID = rs.Field("paymentid").StringValue
//Add it to our list.
m_oBillingArray.Append oEntry
else
oEntry = new clsHistory
oEntry.sPaymentID = rsPaymentApplications.Field("payment
tempDate.SQLDateTime = rsPaymentApplications.Field("tsda
oEntry.dblDate = tempDate.totalSeconds

oEntry.sDate = tempDate.ShortDate + " " + tempDate.Short
if rs.Field("PaymentType").StringValue <> "" then
oEntry.sType = rs.Field("PaymentType").StringValue
else
oEntry.sType = "Payment"
end if
if rs.Field("SourceType").StringValue = "Refund" then
oEntry.sType = "Refund"
end

//Find any payments
s = "SELECT account, sRef FROM Payment WHERE PaymentID =
if m_bKillPopulateBillingThread OR m_bWeAreClosing OR g
```

## Circular References

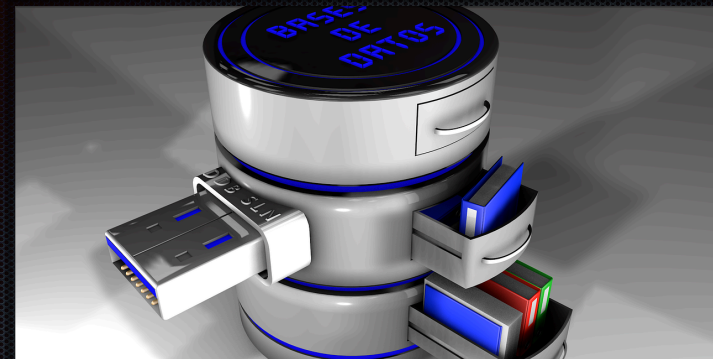
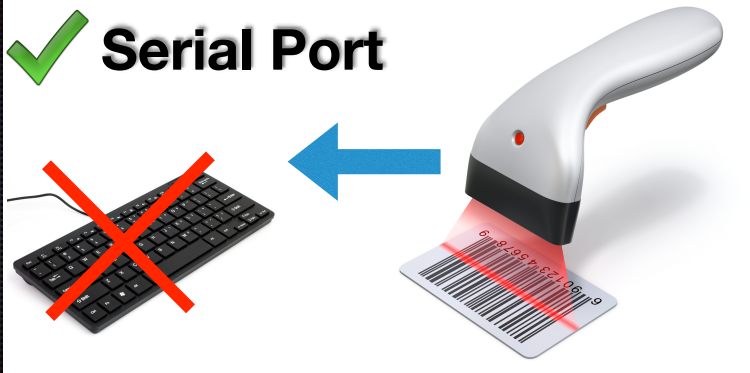


- Destructors: make sure references are nulled
- Use WeakRef instead



## Serial Devices as Keyboard Device

✓ Serial Port



## Database Mistakes

"We build our computer (systems) the way we build our cities: over time, without a plan, on top of ruins." - Ellen Ullman

## Recordset <> nil No DB Error Checking

```
' Load the Data
rs = g_DB.SQLSelect(SQL)
if rs <> nil then
    while not rs.EOF
        if m_sVLIIDs <> "" then m_sVLIIDs = m_sVLIIDs + ","
        m_sVLIIDs = m_sVLIIDs + rs.Field("id").StringValue

        rs.MoveNext
    wend
    rs.Close
end if
```

## Recordset <> nil No DB Error Checking (2)

```
' Load the Data
rs = g_DB.SQLSelect(SQL)
If g_DB.error Then
    DBErrMsg g_DB.ErrorCode, g_DB.ErrorMessage, SQL
    Return
EndIf...
While Not rs.EOF
    If m_sVLIIDs <> "" Then m_sVLIIDs = m_sVLIIDs + ","
    m_sVLIIDs = m_sVLIIDs + rs.Field("id").StringValue

    rs.MoveNext
Wend
rs.Close
```



## Not Using Prepared Statements

- The “O’Neill” Factor in SQL statements
- Formatting for other data types
- PreparedStatements
  - Prevents SQL Injection Attacks
  - Automatically convert data formats for you
  - No SQL String formatting!
  - SAFER - not much more work

## Not Using Transactions

```
if Data.TableExists("t_dbversion",db) = true then return
db.SQLExecute( "BEGIN TRANSACTION" )
db.SQLExecuteRaiseOnError kCreateVersionTable
db.Commit
```

- Temporarily suspends DB Writes
- Creates a temporary state that can be Rolled Back
- Great for a burst of DB operations
- Best way for Parent-Child record Inserts, Updates, or Deletes
- Only way to properly recover from a DB Error

## Looping Recordset

```
rs = g_DB.SQLSelect("SELECT * FROM aTable WHERE aTable_id = 1;")
//Check for error
While rs.eof = False
  Dim FieldData As String = rs.field("someField").StringValue

  Dim sql2 As String = "Select * from bTable aWhere bTable_id = " + fieldData
  Dim rs2 As RecordSet = g_db.SQLSelect(SQL2)
  //check for error
  While rs2.eof = False
    Dim sData As String = rs2.Field("someField2").StringValue

    rs.edit
    rs.Field("bTableData").StringValue = sData
    rs.Update
    //Check for error

    rs2.eof = False
  Wend

  rs.MoveNext
Wend
```

## Not Using DB Views

- Looping Recordset to get Data from related tables
- Use a database View to Join the data
- Combines only the data you need
  - No “Get \* from TableA and TableB”
- Drawbacks
  - Xojo views not updatable
  - Sometimes slow
  - Not always easy to create. GUI editors helpful



# Not Enough Test Data

- Development Phase
  - Create limited data to prove it works
  - No thought on amount of Data
- Production Phase
  - Client adds millions of rows of data in multiple tables
  - Slow queries/inserts/updates
- Solution
  - Index fields that are queried often
  - Limit usage of Likes in where clauses
  - Use Transactions

# Adding FieldNameX to Accommodate Additional Data

TABLES	Name	Type
sqlite_master	User_ID	INTEGER
sqlite_sequence	Name	TEXT
<b>Users</b>	Email	TEXT
	Phone	TEXT
OPTIONS	Address	TEXT
main	Phone2	TEXT
	Phone3	TEXT
	Email2	TEXT
	Email3	TEXT

# 'Combined' Data in a Single Field

Users	Email	TEXT
-------	-------	------

```
{
  "Main_Email" : "bobk@bkeeney.com",
  "Private_Email" : "bobk@example.com",
  "Support_Email" : "support@bkeeney.com"
}
```


bobk@bkeeney.com, bobk@example.com, support@bkeeney.com

# Normalized Email Table


Name	Type
Email_ID	int
EmailType_ID	int
EmailAddress	text
EmailPrimary	tinvent
AssociationType_ID	int
AssociatedRecord_ID	int
Inactive	tinvent
CreatedBy	int
CreatedDate	datetime
ModifiedBy	int
ModifiedDate	datetime



## Human Readable/ Changeable Primary Keys



Name	Type	Con
Name	TEXT	P
Email	TEXT	P
Phone	TEXT	P
Address	TEXT	P



Name	Email	Phone	Address
Sue Smith	sues@example...	555-1212	1212 Main
Jane Doe	janed@exampl...	123-4567	7010 Sussex
Sue Jones	suej@example....	556-5555	321687 Ranger

## Managing your Own Record ID's

- Don't do it!
- You are not smarter than Database
- Let the DB do what it's good at!
- Never a good reason to do it.



## Miscellaneous Design Mistakes

"The first step of any project is to grossly underestimate its complexity and difficulty." - Nicoll Hunt

## Not Using Container Controls

- Putting 100's of controls on a window is slow
- Use container controls to break up UI logically
  - IDE creates a composite picture for container
  - Speeds it up
- Break up CODE logically as well
  - Pass object into object
  - Implement events for parent



## Buying Encrypted Source Code



- Sure, it's cheaper
- But:
  - Totally dependent on developer to
    - Maintain
    - Answer simple questions
- No long term advantage to Encrypted Source Code

## Not Code Signing Your Applications



- Code Signing is Required for Mac/Windows Stores
- Let's your app be 'known'
- MacOS
  - Default behavior to block unknown apps from running
  - Need Apple Developer Account \$99/year
- Windows
  - UAC dialog shows "unknown" developer
  - We use K Software. \$84/year
  - We sign all files AND the installer/uninstaller

## No Source Code Management/Backup



- Use the Xojo project format (not binary)
- Git and Subversion
  - Track changes down to the line of code
- Commit early and often
  - Minimum of twice a day
  - When any major task is complete (not debugged)
  - Use reasonable commit comments!
- Track releases and milestones
- Offsite is better!

## Not Testing on ALL Target Platforms Early and Often



- Develop on preferred platform
- Test first and often on preferred platform
- Other targets are afterthoughts
- Major UI and App characteristic issues



## Final Thought



“As a software developer, you are your own worst enemy. The sooner you realize that, the better off you’ll be.”

- Jeff Atwood

## Q & A



Bob Keeney

[bobk@bkeeney.com](mailto:bobk@bkeeney.com)

Give us feedback on this session in the XDC app!