Rapid Database Development

Bob Gordon
Salud Systems

Programmers are Lazy*

If you are like me, you don't want to think too much about what you're doing.

And, it would be best we could work on different platforms in the same way.

* Try searching "programmers are lazy" for citations.

The Typical Database Application

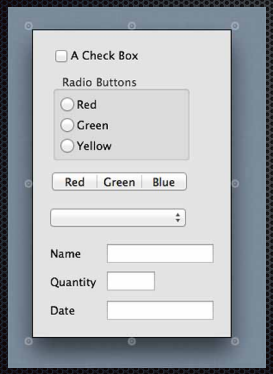
Users Want/Need To

- Enter Data
- Locate Stuff Already Entered
- Make Some Reports

And do this reliably and efficiently

Xojo Controls

Xojo provides a nice collection of controls



However, they have different "natural" values.

What to Store?

Unless you can do arithmetic on the value, store it as text.

- Might take up more room.
- However, in most situations these days, space does not matter.
- The data is easier to read (is “3” the code for “red?”).
- Database engines sometimes deal with data types differently (e.g. Boolean).
- Never store Booleans.

Make Everything On the Screen Text

A text field is always text.

If we make all the other controls understand text we can do something like:

```
aControl.setValue(value as string)
```

and

```
value = aControl.getValue()
```

Make All Controls the Same

While we’re at it we can add some functionality:

isEmpty()

isRequired()

setLabel()

Et cetera

RDDControl Class Interface

Every control that accepts data conforms to the same interface.

As far as the rest of the program is concerned, we have only one type of control.

Note: All the RDD stuff is prefixed with “rdd.” I would have put it in a namespace, but namespaces cannot include Windows or Containers.

How This Works

Xojo windows and container controls maintain a list of all the controls they contain.

We simply loop over all the controls doing what is necessary.

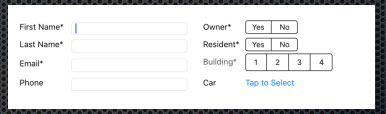
Any control that accepts data is in a container.

A container only deals with a single table.

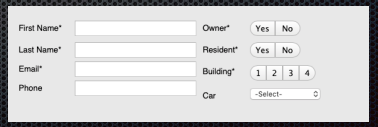
What This Looks Like

Desk

iPad



Web



Controls I Avoid

- Radio Buttons
  - Take up too much room
- Check Boxes
  - Can't tell if there's no entry or it's been missed

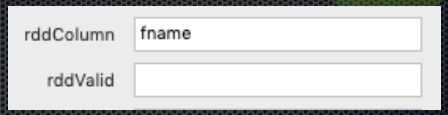
Connecting Things

A control needs to know:

- Its database column
- Validation
- Its label

Column and Validation

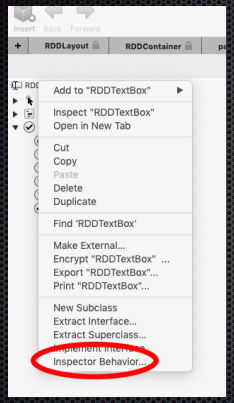
Add two public propereties to each RDDControl.

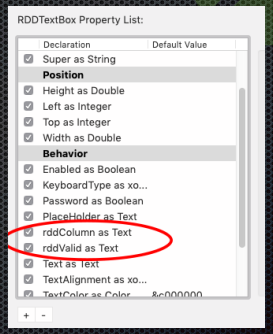


rddColumn holds the name of the column.

rddValid holds the name of the validation to use (if any) e.g. email, phone, state, zip, etc. There's a class that knows rules.

Nifty Xojo Feature





Getting the Label

Each control may have a related label.

Identified by a naming convention:

txtFname

lblFname

I could probably use ctl as the prefix for all the RDDControls.

```
/// class:       RDDContainer
/// method:      findLabels
/// summary:     links each control to its label
/// parameters:  none
/// returns:     none
/// author:      Bob Gordon
/// date:        March 12, 2019
/// note:        This works because of the naming convention.
///              All control names start with a three-character
///              prefix and all labels start with lbl.

dim x1i    as integer
dim x1j    as integer
dim ioc    as iosControl
dim labels as Dictionary
dim name   as Text

labels = new Dictionary()

x1j = self.ControlCount - 1

// first collect all the labels
for x1i = 0 to x1j
  ioc = self.control(x1i)

  if ioc isa RDDLabel then
    dim rdl as RDDLabel
    rdl = RDDLabel(ioc)
    name = rdl.name
    labels.value(rdl.Name) = rdl
  end
next

// loop through the controls and
// see if there's a related label
for x1i = 0 to x1j
  ioc = self.control(x1i)

  if ioc isa RDDControl then
    dim rdc as RDDControl
    rdc  = RDDControl(ioc)
    name = "lbl" + rdc.getName.Mid(3)
    if labels.HasKey(name) then
      rdc.setLabel(labels.value(name))
    end

    If rdc IsA RDDTechBox Then
      RDDTechBox(rdc).showTech(False)
    end
  end
next
```

Since a control knows its label, validation can easily highlight it if there's a problem.

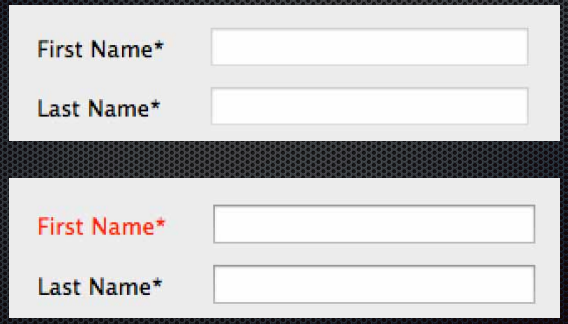
Validation

Three types of validation:

- An entry is required
- An entry is correct (a phone number looks like a phone number)
- An entry is correct depending on another entry (the end date can not be before the start date)

Required Fields

Users need to know the field is required.



Single Field is Correct

There are common bits of information that show up almost everywhere

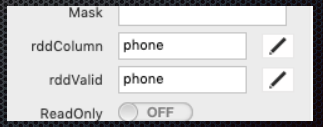
- Phone numbers
- Postal codes
- State/Province Abbreviations
- Et cetera (depending on where you live and the data you're dealing with)

Valid Property

Each control has a valid property.

If it uses one of the standard validations, the validation name is entered.

Currently: phone, date, state, zip, email, password.



Cross Target

Web applications pose a problem:

They basically cannot use any modules or other globals as such things are attached to the application as a whole.

Web apps have a session object, which represents a single user interaction.

Can add properties to the session, which becomes “global” to the session.

The Session Module

To make everything as similar as possible:

- Add a module named “session” to desk and mobile apps.
- Maintains access to database, system information, and app-specific data.
- Now all platforms look the same.

Demo

How does this work in practice?

Let’s build a screen.

Useful Tools

- Source Code Pro
- Keyboard Maestro
- Valentina Studio
- BBEdit

Q & A

Bob Gordon

gordon1bob@yahoo.com