

# **An approach to handling ORDA list boxes**

**Or How I used the hints I got from Miyako**

**Kirk Brooks & Guy Algot**

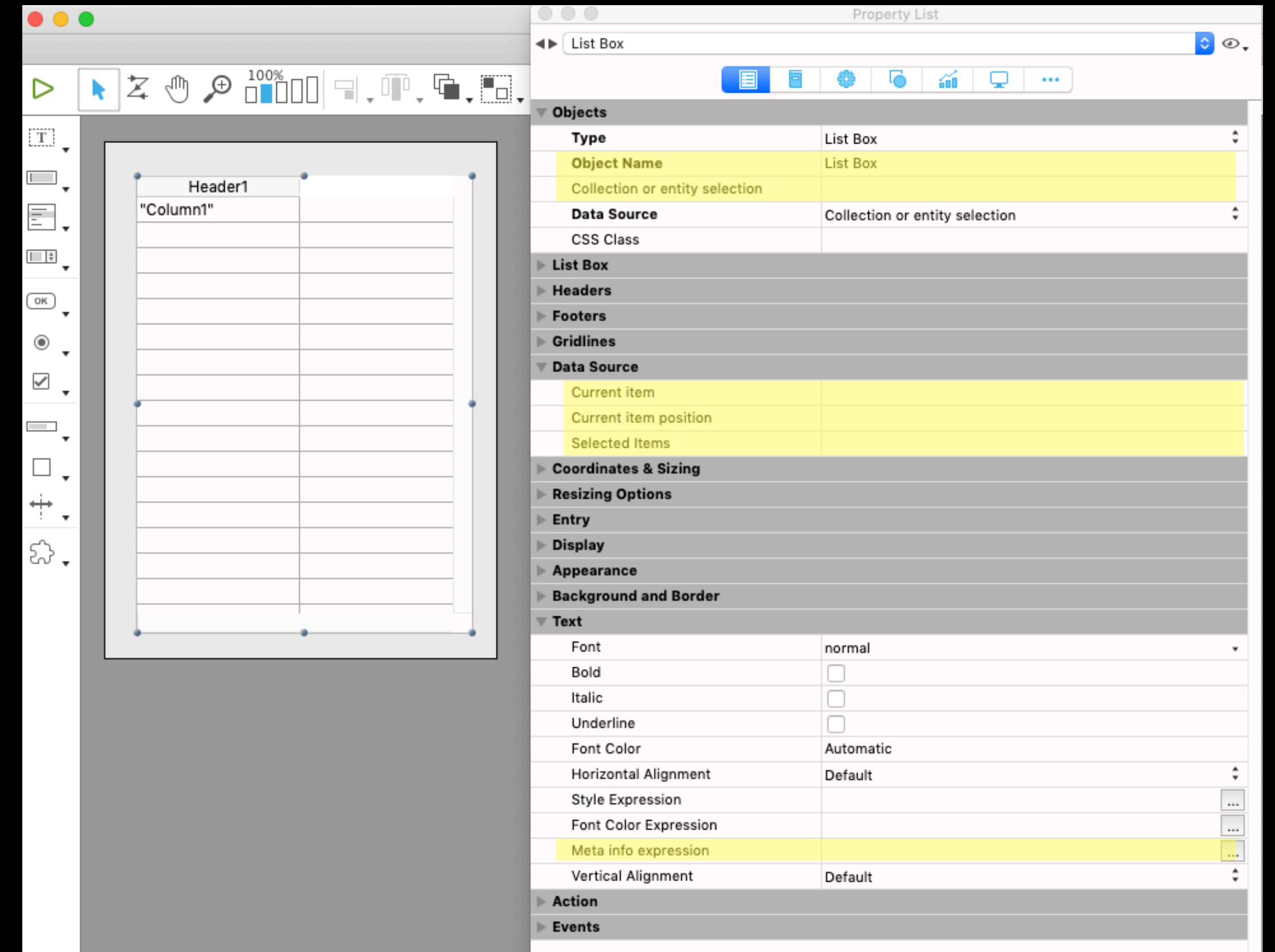
# The Basics

# The Basics

This is focused on collection and entity list boxes.

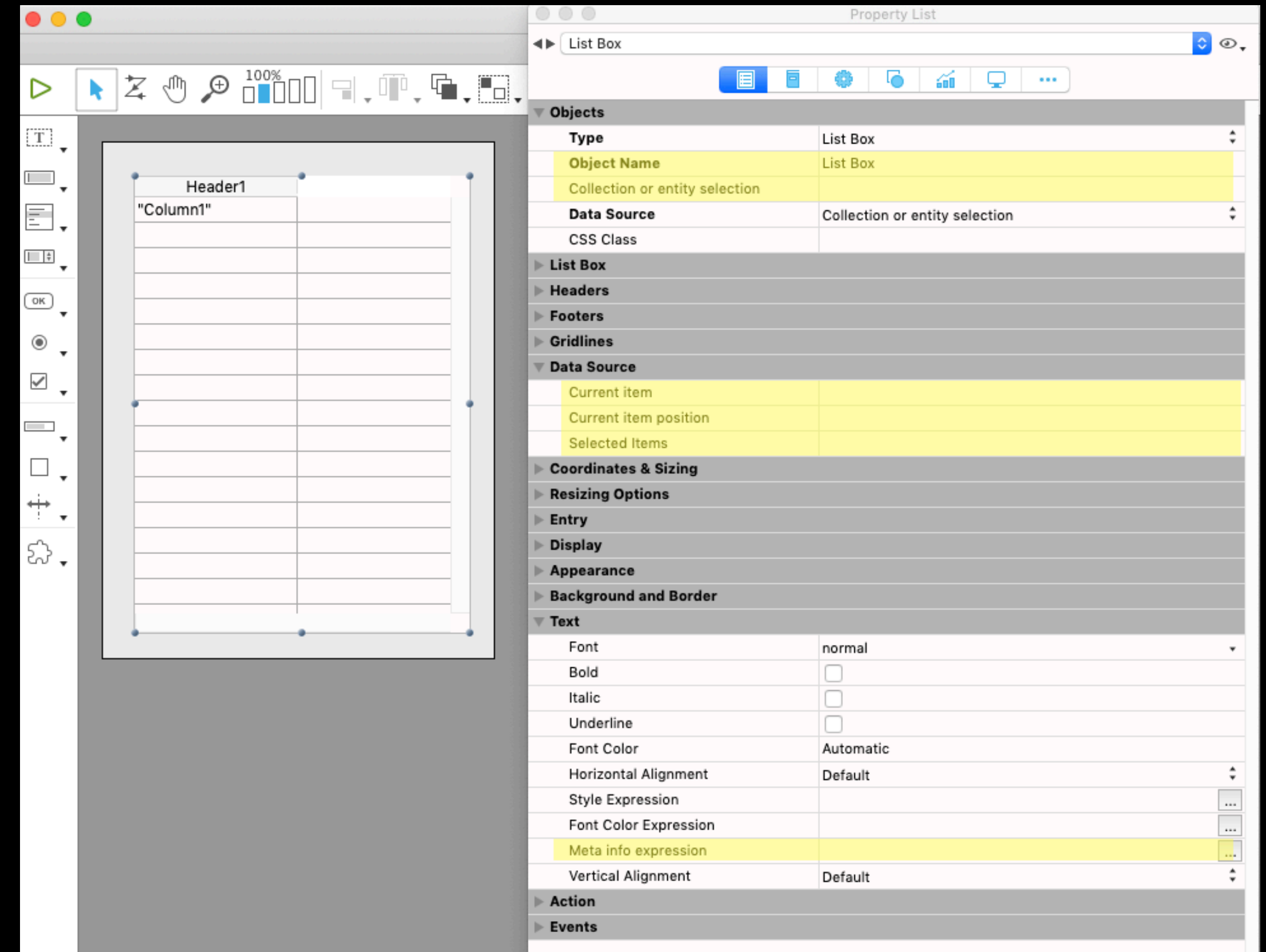
Because I haven't created a new array based listbox in over a year.

# Identifying the parts

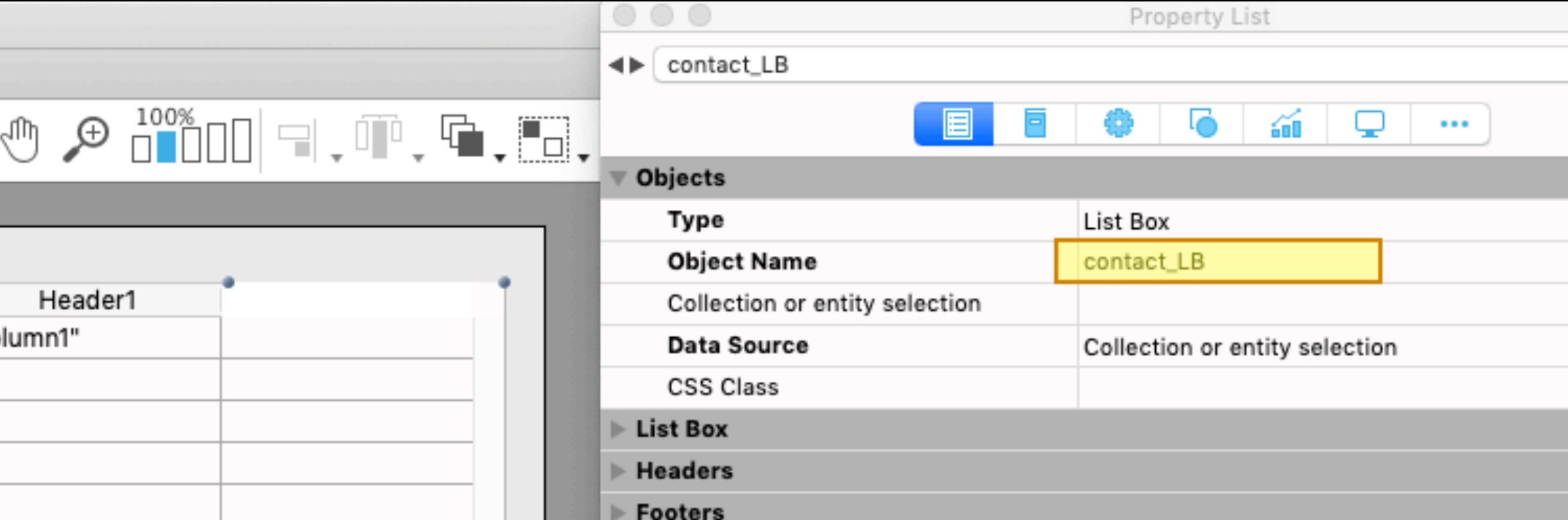


# Identifying the parts

- These 6 properties allow us to identify and control the list box
- The key is to have a scheme we can remember and extends well

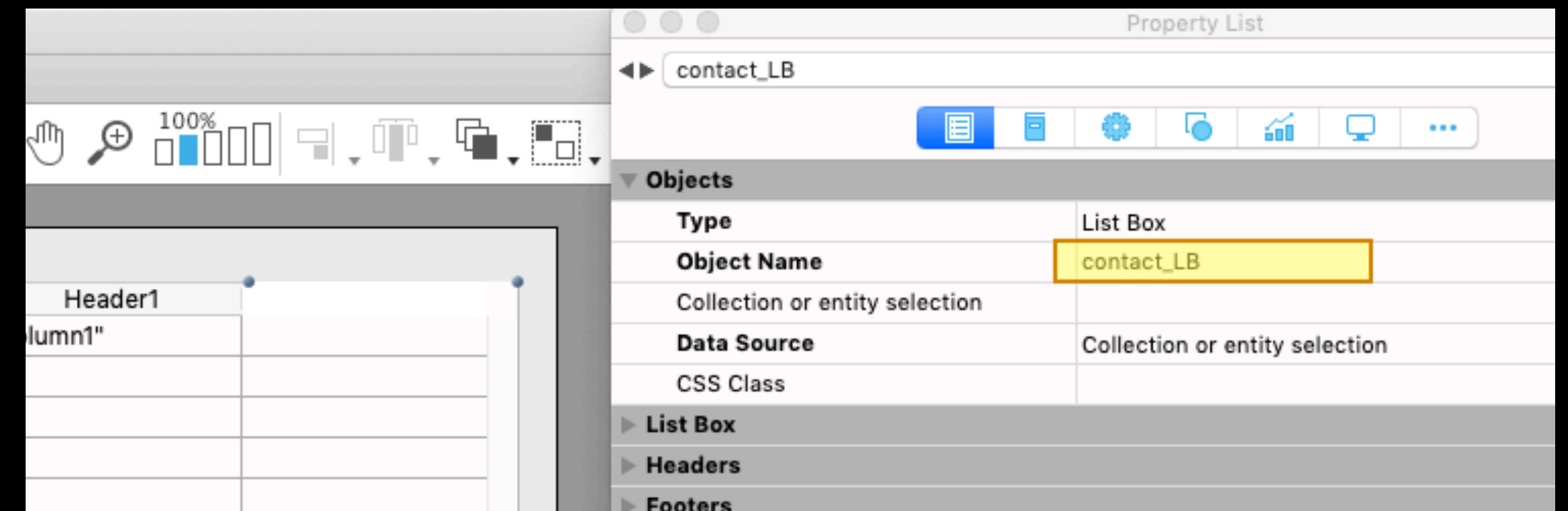


# Choose a name



# Choose a name

- Begin or end with “LB” or whatever you use to indicate a list box
- Make it distinctive
- This will become the “namespace” for this list box



# Initialize the list box / namespace

```
▼ Case of
  ▼ : (Form event code=On Load)

    Form.contact_LB:=New object("name";"contact_LB")
    Form.contact_LB.data:=Null // collection or entity selection
    Form.contact_LB.curItem:=Null // currently selected item
    Form.contact_LB.pos:=Null // position (not index) of current item
    Form.contact_LB.selected:=Null
    Form.contact_LB.meta:=New object // meta expression object |

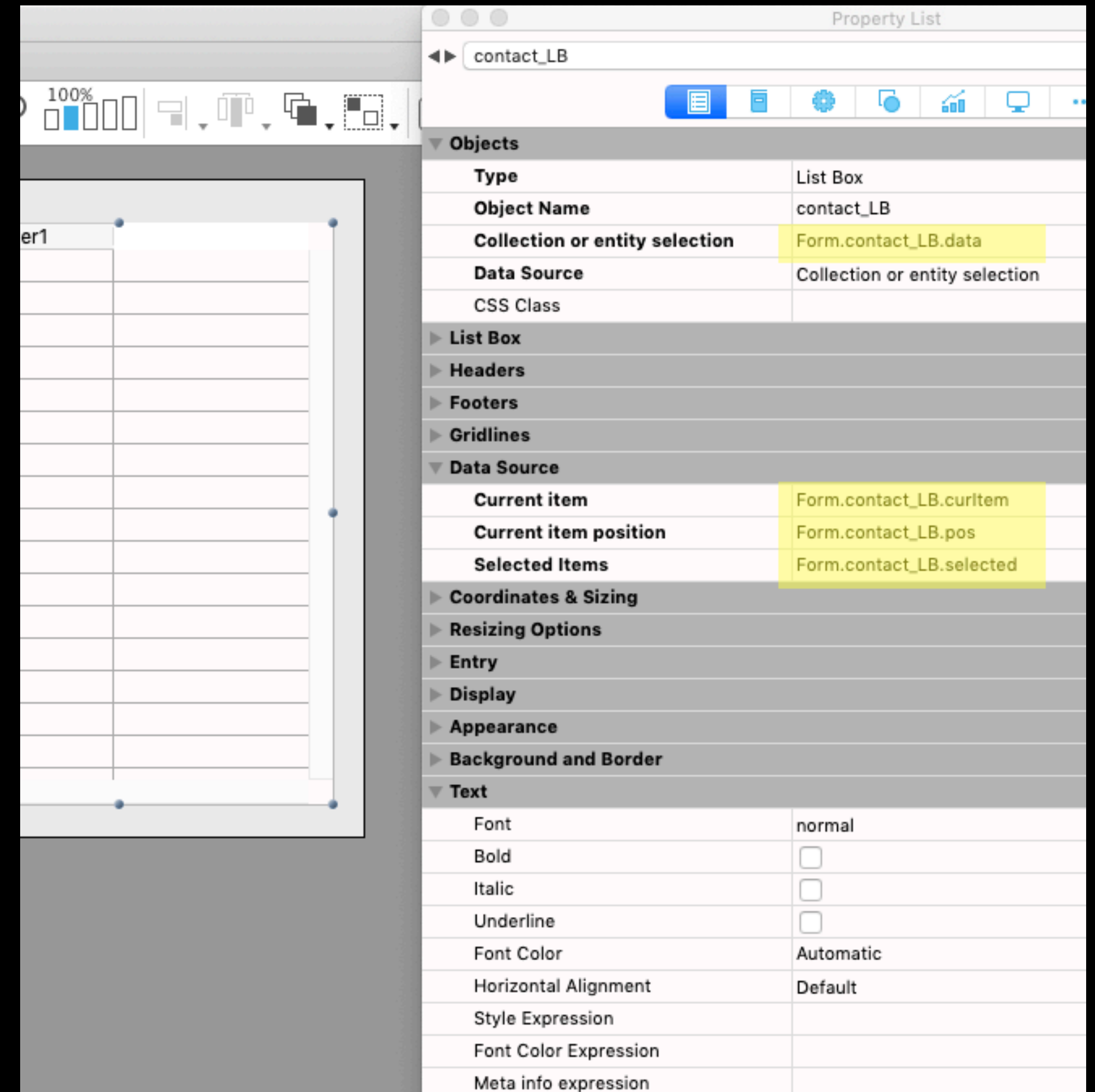
  End case
```



**Apply these references to the list box**

# Apply these references to the list box

- We have a consistent naming scheme based on the object name
- Very helpful when there are multiple list boxes on the form
- And we could move this to a method...



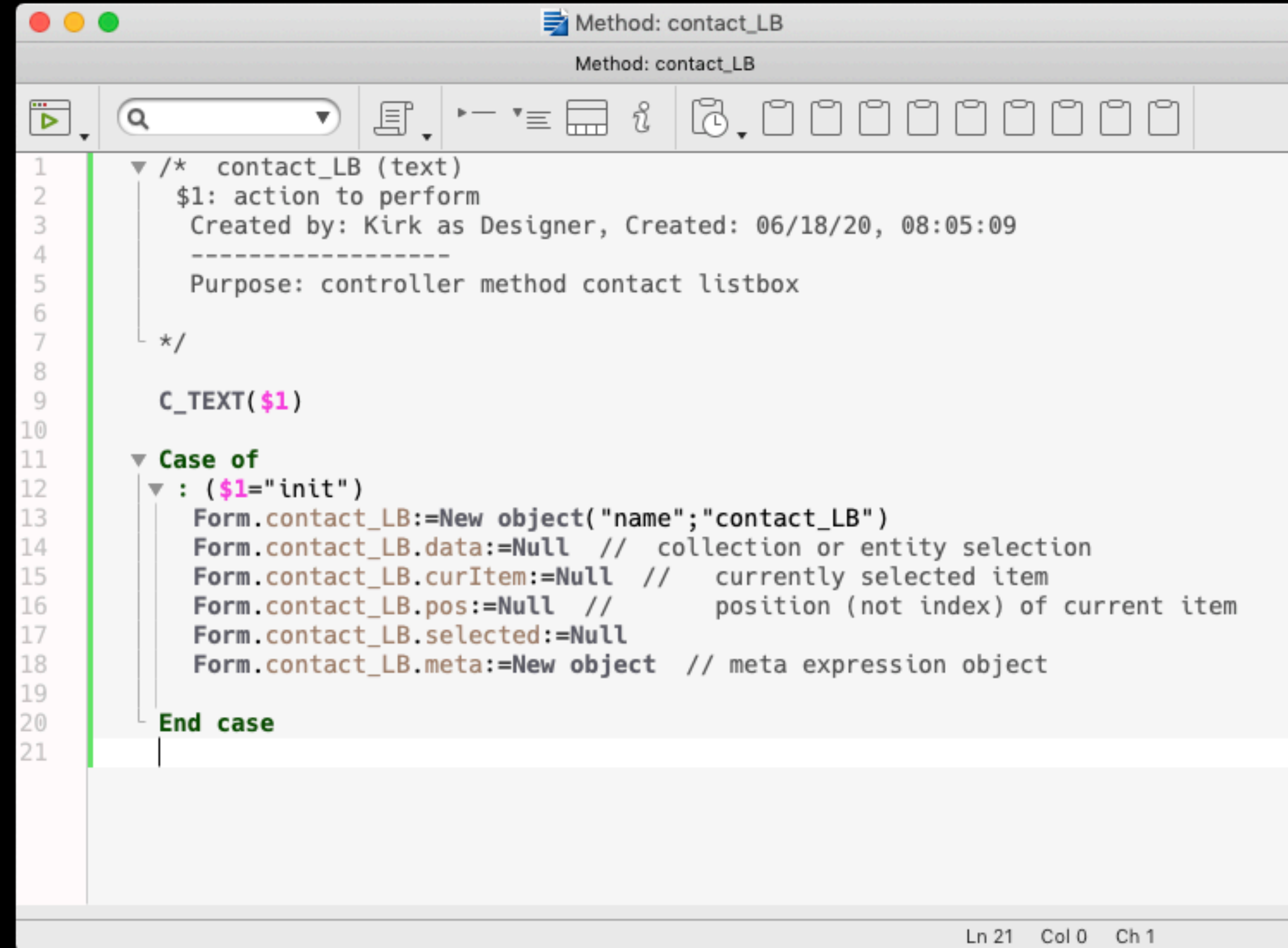
The screenshot displays a software interface with a list box on the left and a 'Property List' panel on the right. The list box is titled 'er1' and contains several empty rows. The 'Property List' panel is titled 'contact\_LB' and shows various properties for the list box. The 'Data Source' section is expanded, showing properties like 'Current item', 'Current item position', and 'Selected Items', all with values starting with 'Form.contact\_LB.'. The 'Text' section is also expanded, showing properties like 'Font', 'Bold', 'Italic', 'Underline', 'Font Color', 'Horizontal Alignment', 'Style Expression', 'Font Color Expression', and 'Meta info expression'.

Property List	
contact_LB	
100%	
er1	
▼ Objects	
Type	List Box
Object Name	contact_LB
Collection or entity selection	Form.contact_LB.data
Data Source	Collection or entity selection
CSS Class	
► List Box	
► Headers	
► Footers	
► Gridlines	
▼ Data Source	
Current item	Form.contact_LB.curItem
Current item position	Form.contact_LB.pos
Selected Items	Form.contact_LB.selected
► Coordinates & Sizing	
► Resizing Options	
► Entry	
► Display	
► Appearance	
► Background and Border	
▼ Text	
Font	normal
Bold	<input type="checkbox"/>
Italic	<input type="checkbox"/>
Underline	<input type="checkbox"/>
Font Color	Automatic
Horizontal Alignment	Default
Style Expression	
Font Color Expression	
Meta info expression	



# A controller method

- Name the method with our namespace name
- Group the various actions and code the list performs
- And we start to see a number of similarities...



The screenshot shows a code editor window titled 'Method: contact\_LB'. The code is written in a language that uses comments for documentation and specific syntax for object creation and property assignment. The code is as follows:

```
1  /* contact_LB (text)
2     $1: action to perform
3     Created by: Kirk as Designer, Created: 06/18/20, 08:05:09
4     -----
5     Purpose: controller method contact listbox
6
7  */
8
9  C_TEXT($1)
10
11  Case of
12  : ($1="init")
13      Form.contact_LB:=New object("name";"contact_LB")
14      Form.contact_LB.data:=Null // collection or entity selection
15      Form.contact_LB.curItem:=Null // currently selected item
16      Form.contact_LB.pos:=Null // position (not index) of current item
17      Form.contact_LB.selected:=Null
18      Form.contact_LB.meta:=New object // meta expression object
19
20  End case
21
```

The editor interface includes a search bar, a toolbar with various icons, and a status bar at the bottom indicating 'Ln 21 Col 0 Ch 1'.

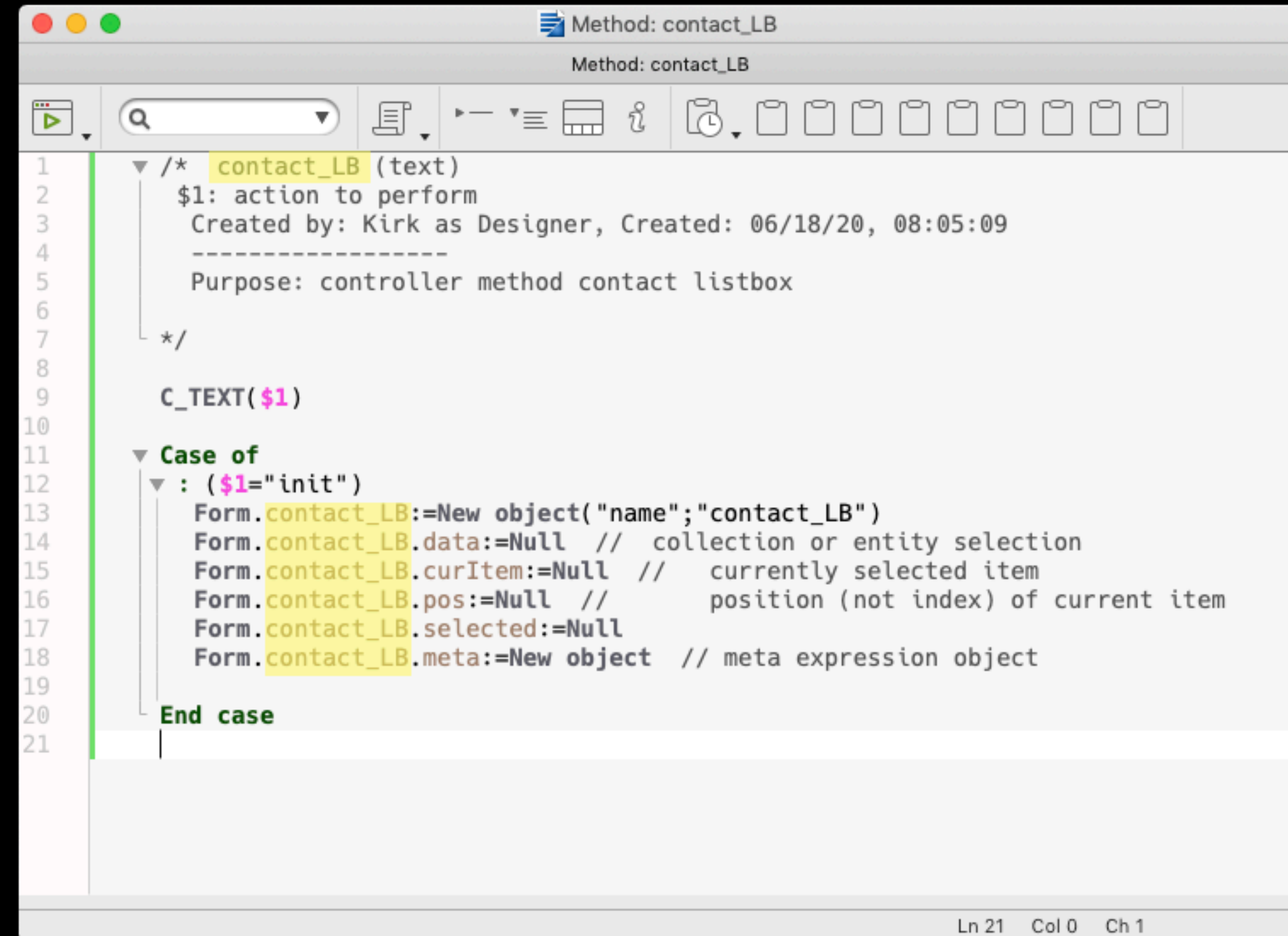
# A controller method

# A controller method

- It's all the same namespace so why not...

# A controller method

- It's all the same namespace so why not...



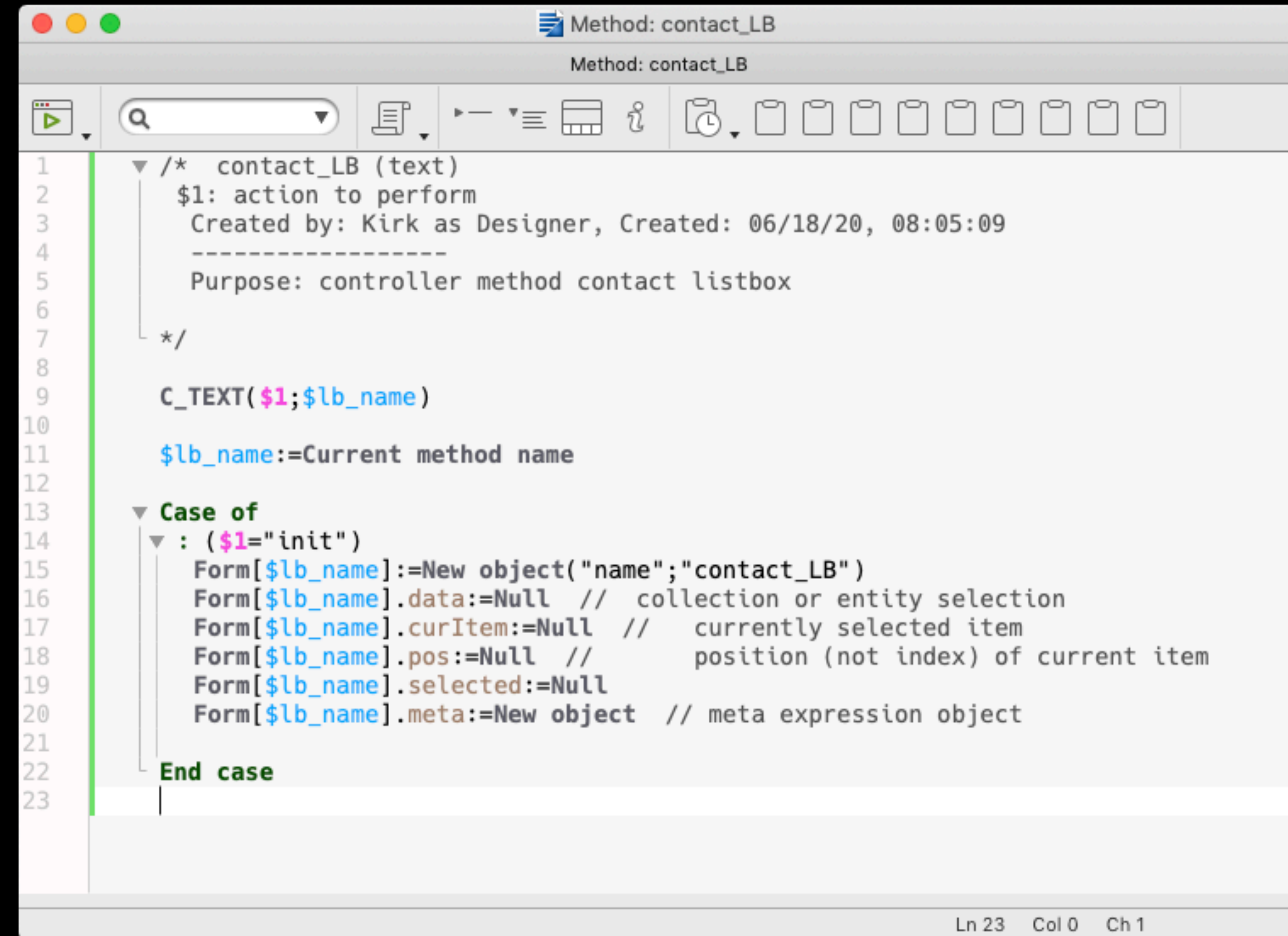
```
1  /* contact_LB (text)
2     $1: action to perform
3     Created by: Kirk as Designer, Created: 06/18/20, 08:05:09
4     -----
5     Purpose: controller method contact listbox
6
7  */
8
9  C_TEXT($1)
10
11  ▼ Case of
12  ▼ : ($1="init")
13      Form.contact_LB:=New object("name";"contact_LB")
14      Form.contact_LB.data:=Null // collection or entity selection
15      Form.contact_LB.curItem:=Null // currently selected item
16      Form.contact_LB.pos:=Null // position (not index) of current item
17      Form.contact_LB.selected:=Null
18      Form.contact_LB.meta:=New object // meta expression object
19
20  End case
21
```

Ln 21 Col 0 Ch 1

# A controller method



# A controller method

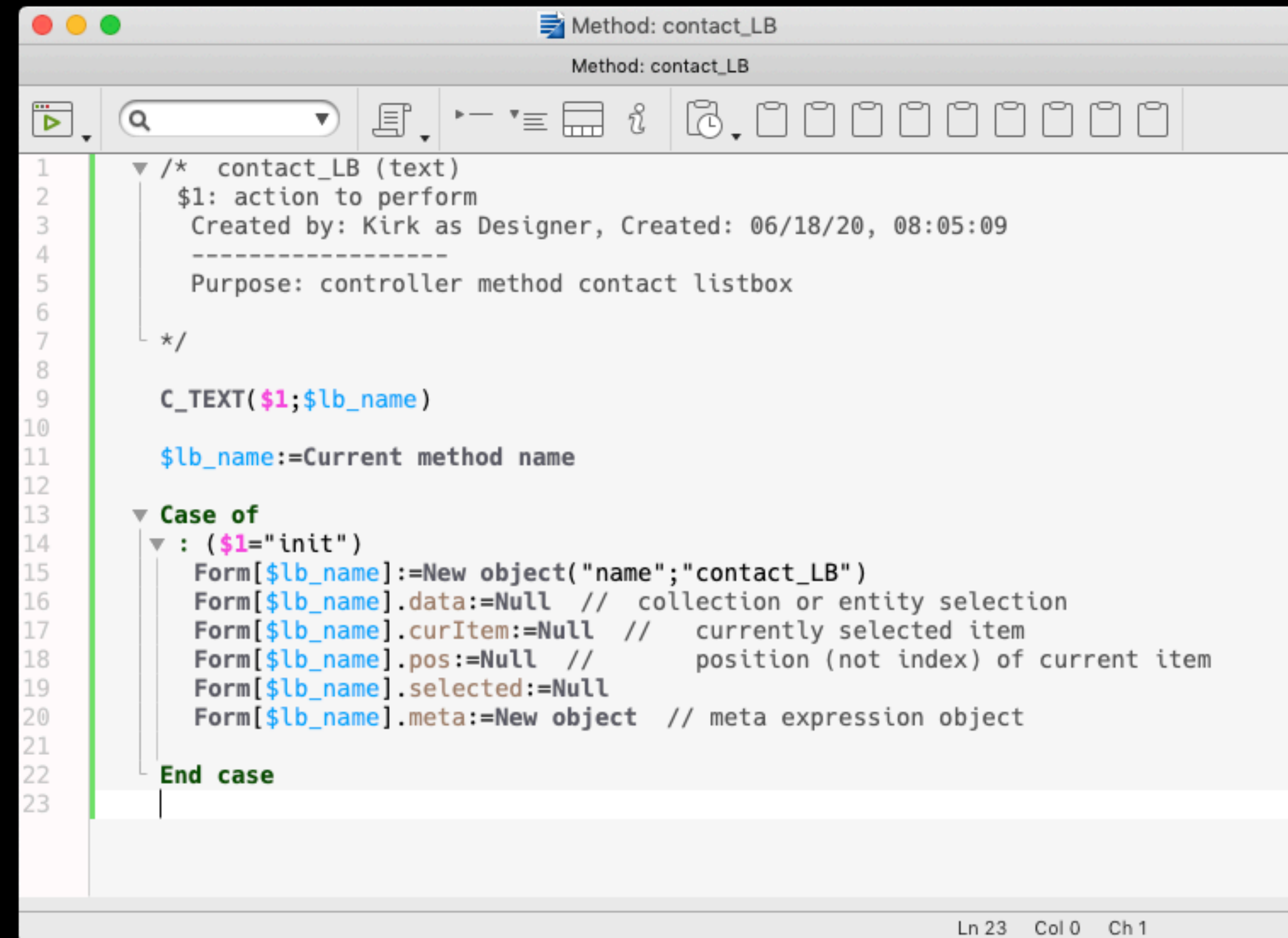


```
1  /* contact_LB (text)
2     $1: action to perform
3     Created by: Kirk as Designer, Created: 06/18/20, 08:05:09
4     -----
5     Purpose: controller method contact listbox
6
7  */
8
9  C_TEXT($1;$lb_name)
10
11  $lb_name:=Current method name
12
13  Case of
14  : ($1="init")
15      Form[$lb_name]:=New object("name";"contact_LB")
16      Form[$lb_name].data:=Null // collection or entity selection
17      Form[$lb_name].curItem:=Null // currently selected item
18      Form[$lb_name].pos:=Null // position (not index) of current item
19      Form[$lb_name].selected:=Null
20      Form[$lb_name].meta:=New object // meta expression object
21
22  End case
23
```

Ln 23 Col 0 Ch 1

# A controller method

- Take advantage of that similarity...



```
1  /* contact_LB (text)
2     $1: action to perform
3     Created by: Kirk as Designer, Created: 06/18/20, 08:05:09
4     -----
5     Purpose: controller method contact listbox
6
7  */
8
9  C_TEXT($1;$lb_name)
10
11  $lb_name:=Current method name
12
13  ▼ Case of
14  ▼ : ($1="init")
15      Form[$lb_name]:=New object("name";"contact_LB")
16      Form[$lb_name].data:=Null // collection or entity selection
17      Form[$lb_name].curItem:=Null // currently selected item
18      Form[$lb_name].pos:=Null // position (not index) of current item
19      Form[$lb_name].selected:=Null
20      Form[$lb_name].meta:=New object // meta expression object
21
22  End case
23
```

Ln 23 Col 0 Ch 1

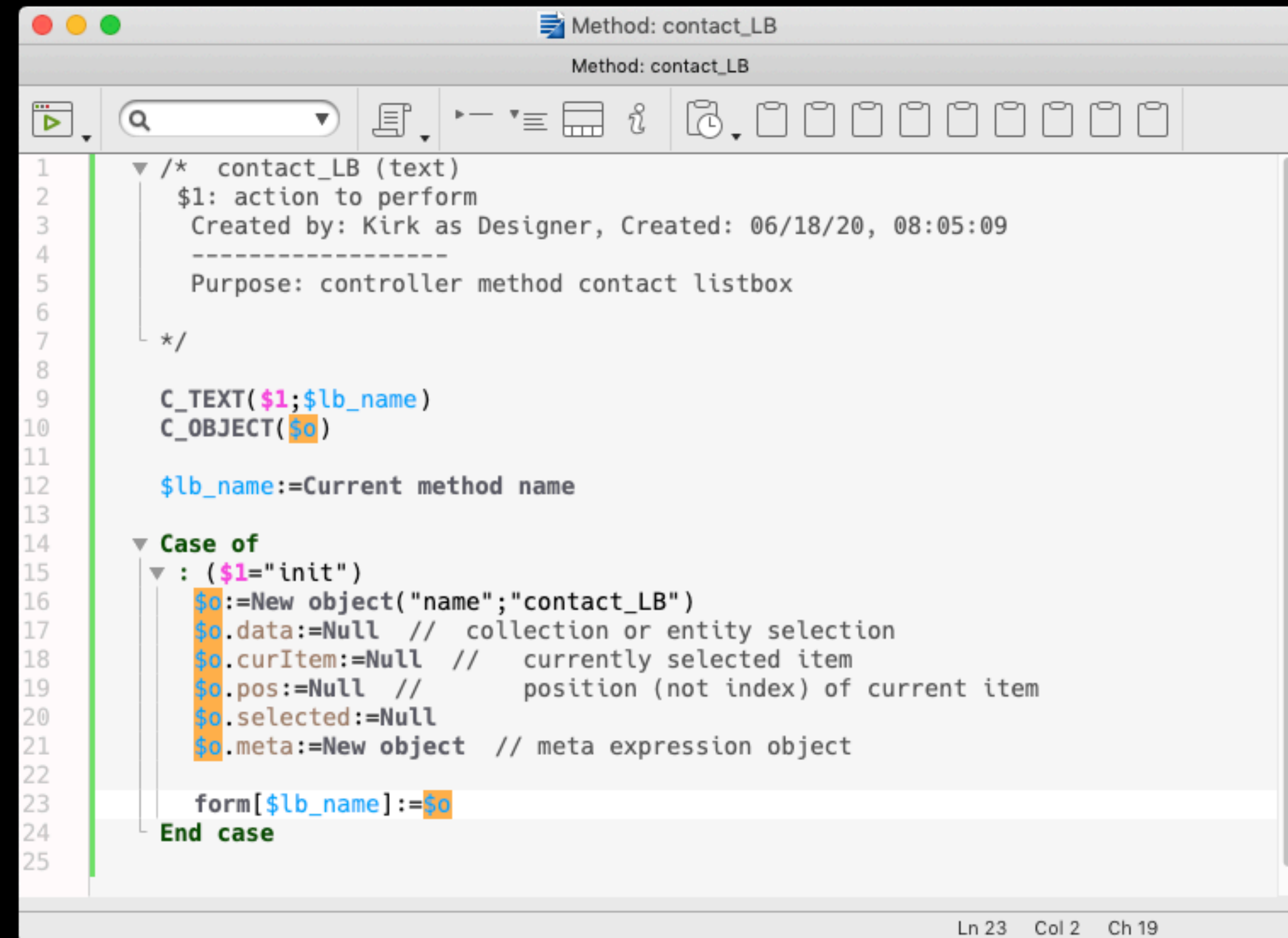
# A controller method

# A controller method

- And save ourselves a lot of typing by using an object which we write to Form as our listbox name

# A controller method

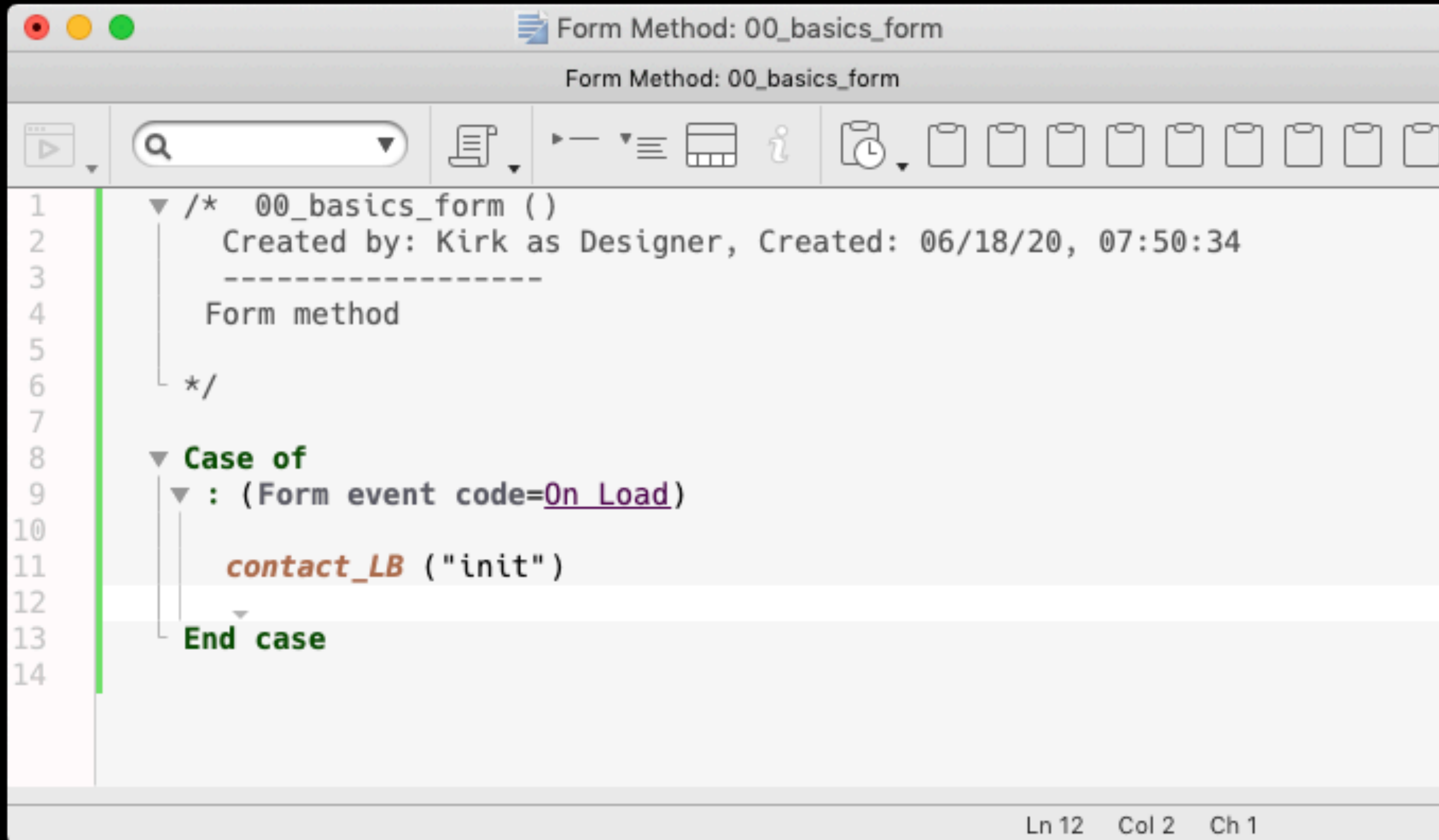
- And save ourselves a lot of typing by using an object which we write to Form as our listbox name



```
1  /* contact_LB (text)
2     $1: action to perform
3     Created by: Kirk as Designer, Created: 06/18/20, 08:05:09
4     -----
5     Purpose: controller method contact listbox
6
7  */
8
9  C_TEXT($1;$lb_name)
10 C_OBJECT($o)
11
12 $lb_name:=Current method name
13
14 ▼ Case of
15   : ($1="init")
16     $o:=New object("name";"contact_LB")
17     $o.data:=Null // collection or entity selection
18     $o.curItem:=Null // currently selected item
19     $o.pos:=Null // position (not index) of current item
20     $o.selected:=Null
21     $o.meta:=New object // meta expression object
22
23     form[$lb_name]:=$o
24   End case
25
```

Ln 23 Col 2 Ch 19

# Now the initialization looks like



The screenshot shows a code editor window titled "Form Method: 00\_basics\_form". The editor displays the following code:

```
1  /* 00_basics_form ()
2     Created by: Kirk as Designer, Created: 06/18/20, 07:50:34
3     -----
4     Form method
5
6  */
7
8  ▼ Case of
9  ▼ : (Form event code=On Load)
10
11     contact_LB ("init")
12
13  End case
14
```

The code is formatted with a light blue background for the comment block and a light orange background for the event code block. The text "contact\_LB" is highlighted in red. The editor includes a search bar, a toolbar with various icons, and a status bar at the bottom showing "Ln 12 Col 2 Ch 1".



# Adding Data

- Simply add a collection or entity selection to the .data property



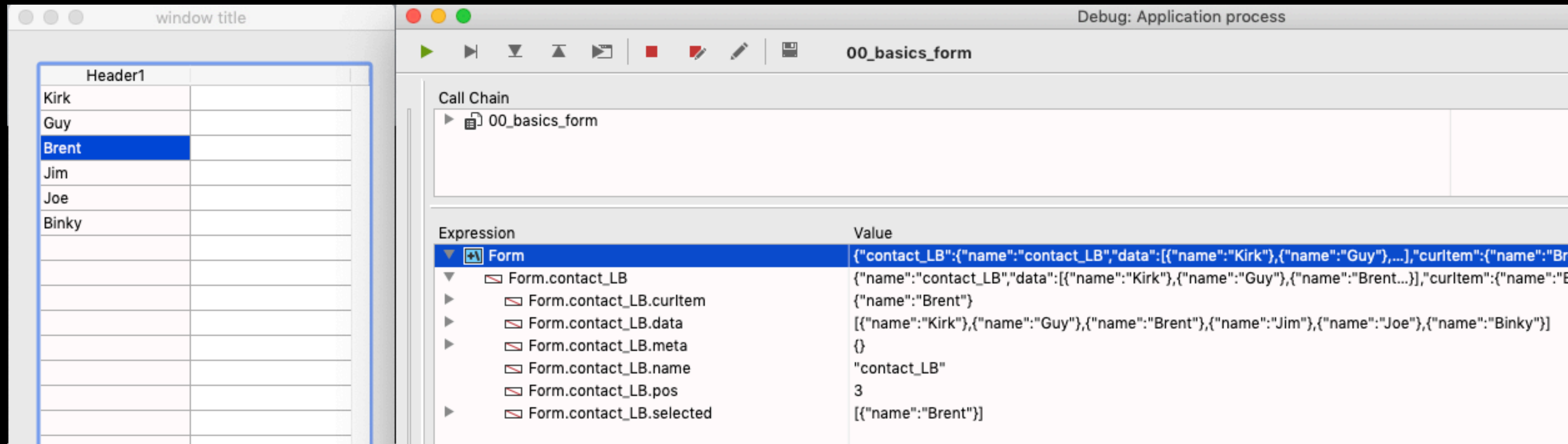
# Let's display a collection



# Let's display a collection

[illegible]

# When we run the form



- pos is position in the listbox, not the index of the collection
- There is no downside to defining all the possible elements

**What about that meta property?**

# What about that meta property?

Which is the hint I got from Miyako

# Meta Info Expression

# Meta Info Expression

- Specifies an expression or a variable which will be evaluated for each row displayed
- It allows defining a whole set of row text attributes
- You must pass an object variable or an expression that returns an object
- We are going to pass an expression, a method, that returns an object

# Meta Info Expression

Style settings made with this property are **ignored** if other style settings are already defined through other expressions

Style Expression

Font Color Expression

Background Color Expression

So how do I actually use this?



# Using the Meta Info Expression

# Using the Meta Info Expression

- Create a collection in the meta property
- Each member of the collection is an object that defines the properties to apply for some condition
- Use [0] as the default expression

# Using the Meta Info Expression

# Using the Meta Info Expression

```
$o.meta:=New collection // meta expression object
$o.meta.push(New object(\
  "stroke";"automatic";\
  "fill";"automatic";\
  "fontStyle";"normal";\
  "fontWeight";"bold";\
  "textDecoration";"normal"))
// [1] will underline the text
$o.meta.push(New object(\
  "stroke";"dodgerblue";\
  "fill";"automatic";\
  "fontStyle";"normal";\
  "fontWeight";"normal";\
  "textDecoration";"underline"))
//[2] will make the text red and italic
$o.meta.push(New object(\
  "stroke";"#ff0000";\
  "fill";"automatic";\
  "fontStyle";"italic";\
  "fontWeight";"normal";\
  "textDecoration";"normal"))
```

- Create the meta collection
- Define the 'default' style
- Define any other styles

# Using the Meta Info Expression

# Using the Meta Info Expression

- We have defined three styles

# Using the Meta Info Expression

- We have defined three styles
- Now we need a method to run and evaluate the row to determine which style to apply

# Using the Meta Info Expression

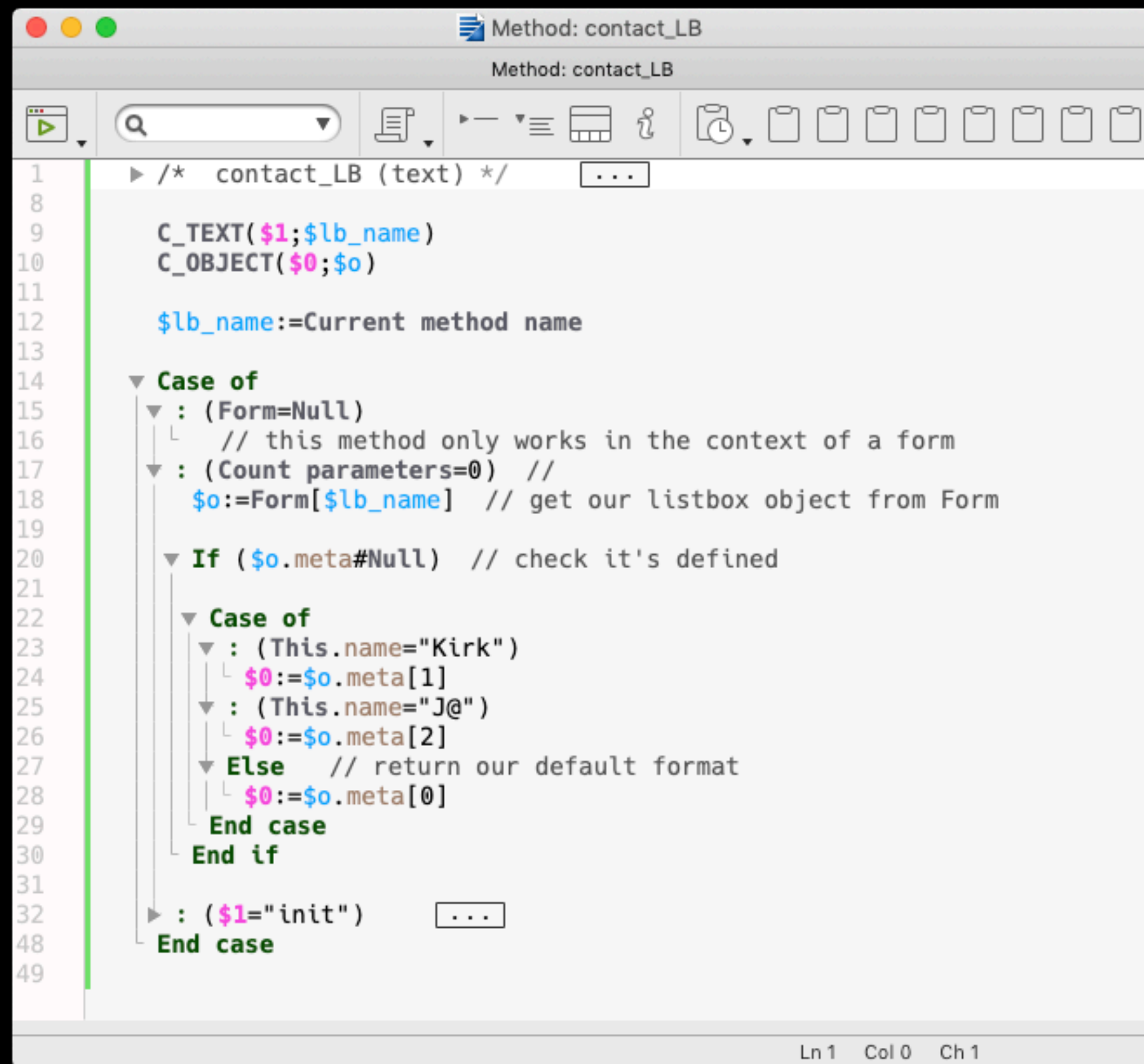
- We have defined three styles
- Now we need a method to run and evaluate the row to determine which style to apply
- Why not use the controller method?



# Using the Meta Info Expression

# Using the Meta Info Expression

- Add a return expression (\$0)
- Add basic error protection
- **This** refers to the current entity or collection member.

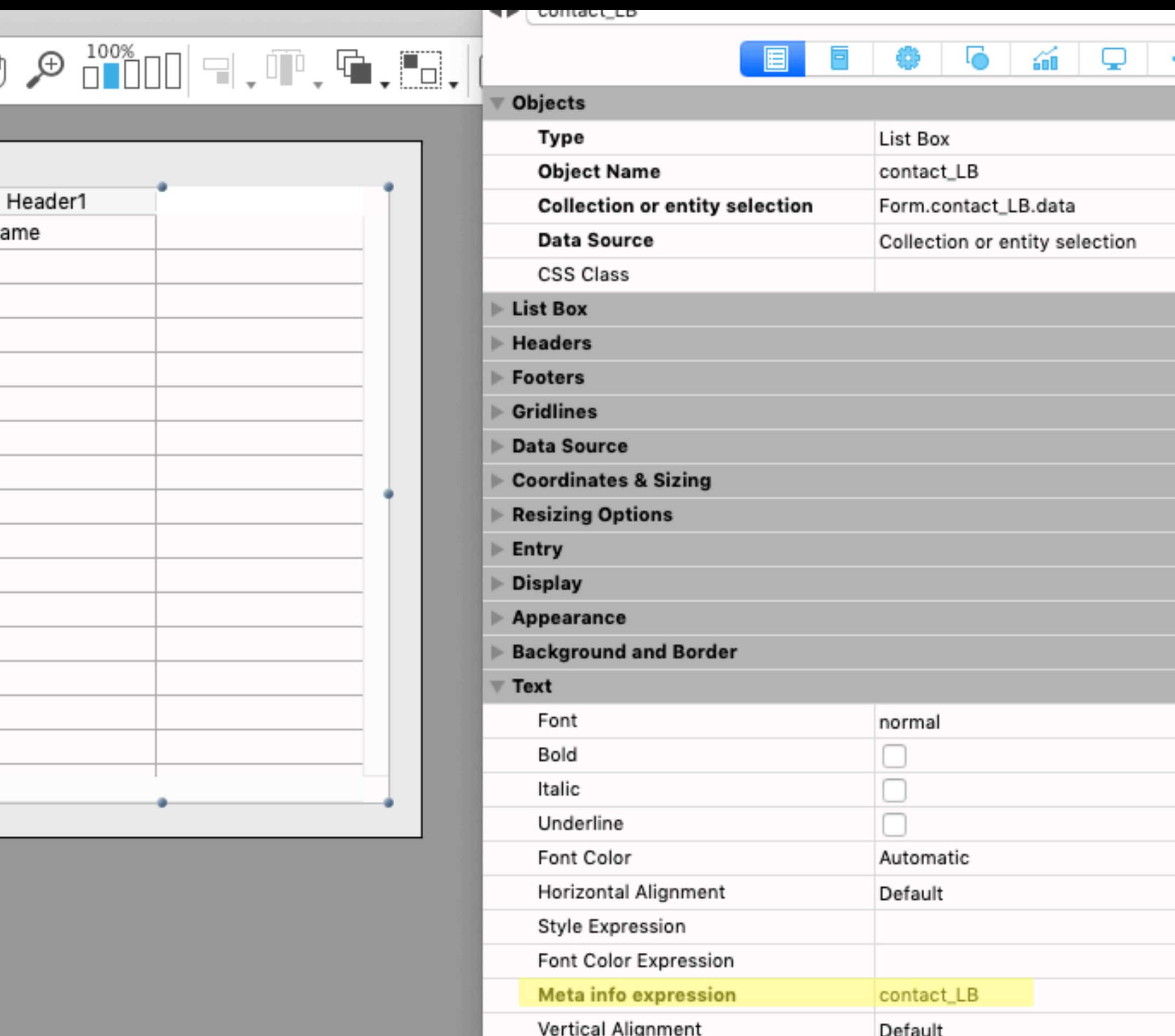


```
1  /* contact_LB (text) */
8
9  C_TEXT($1;$lb_name)
10 C_OBJECT($0;$0)
11
12 $lb_name:=Current method name
13
14 Case of
15 : (Form=NULL)
16   // this method only works in the context of a form
17 : (Count parameters=0) //
18   $0:=Form[$lb_name] // get our listbox object from Form
19
20   If ($0.meta#Null) // check it's defined
21
22     Case of
23     : (This.name="Kirk")
24       $0:=$0.meta[1]
25     : (This.name="J@" )
26       $0:=$0.meta[2]
27     Else // return our default format
28       $0:=$0.meta[0]
29     End case
30   End if
31
32 : ($1="init")
33 End case
48
49
```

The controller method now returns a Meta Info Expression object if called without any parameters...

# Using the Meta Info Expression

# Using the Meta Info Expression



- Add the list box method as the Meta Info Expression
- This will call the method for each row of the list box without any parameters

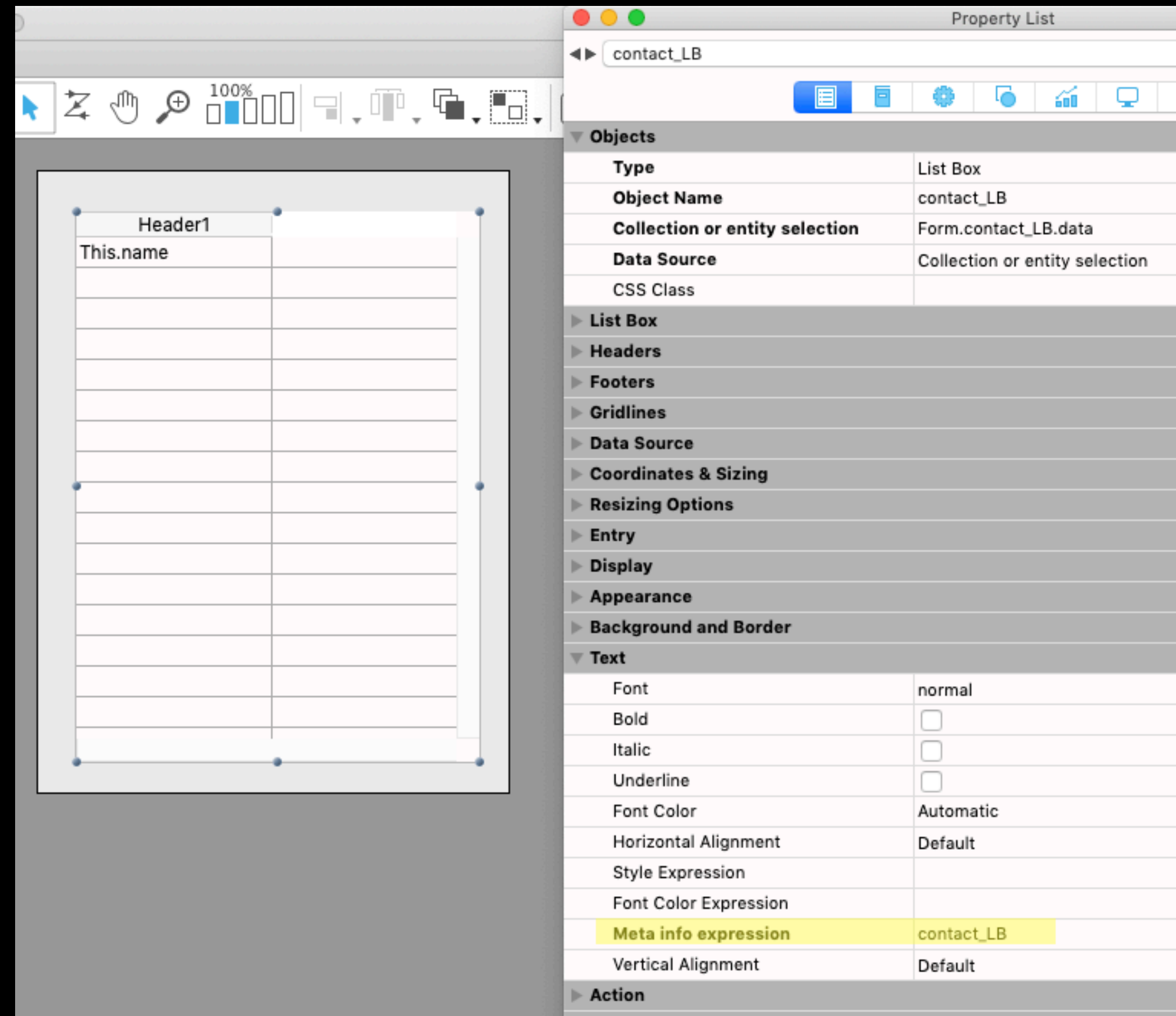
# Using the Meta Info Expression

[illegible]

**A rose is a rose is a rose...**

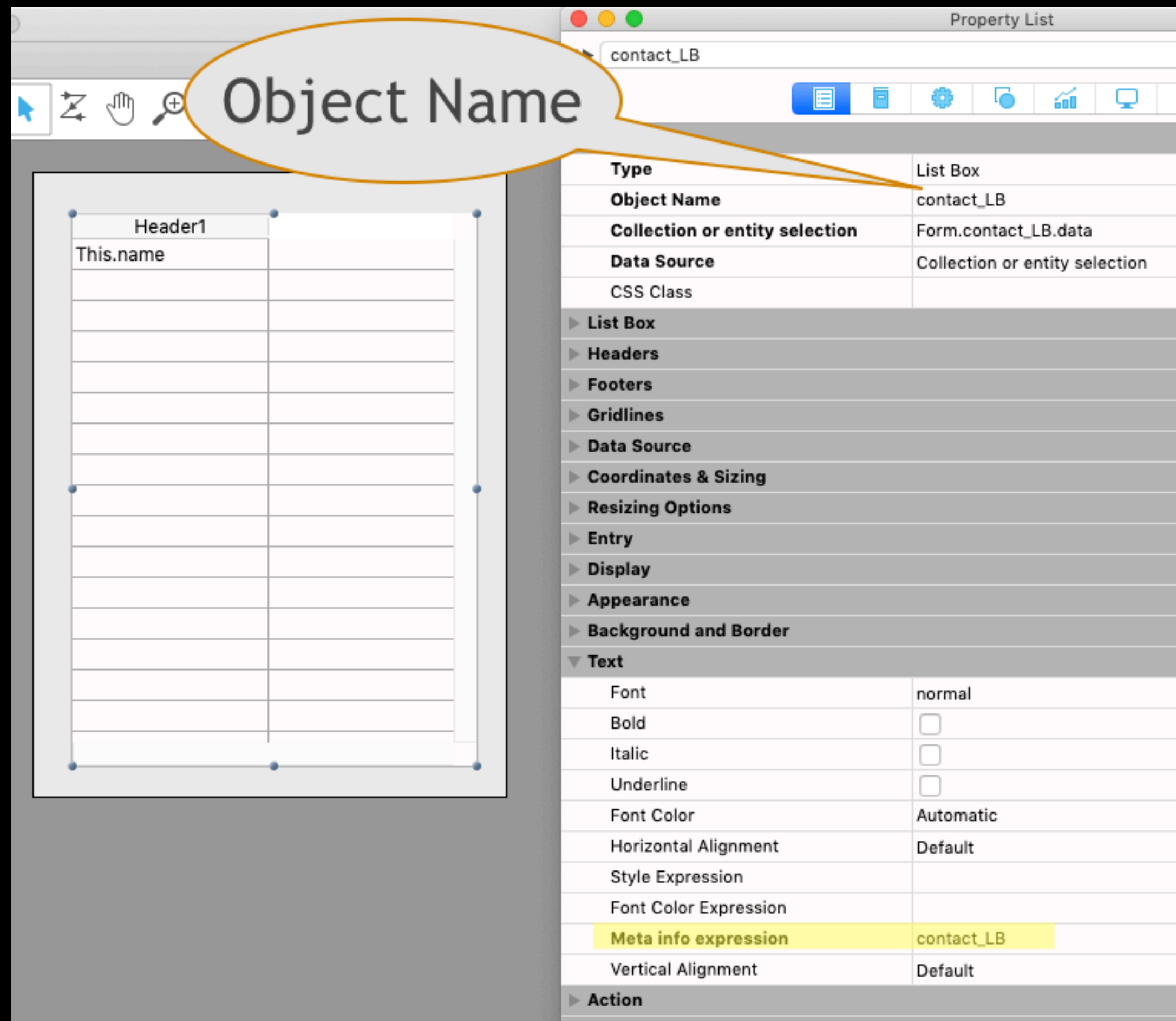
**Because we use the same name doesn't mean it's the same thing**

# Because we use the same name doesn't mean it's the same thing





# Because we use the same name doesn't mean it's the same thing



Because we use the same name doesn't mean it's the same thing

Header1

This.name

Property List

contact\_LB

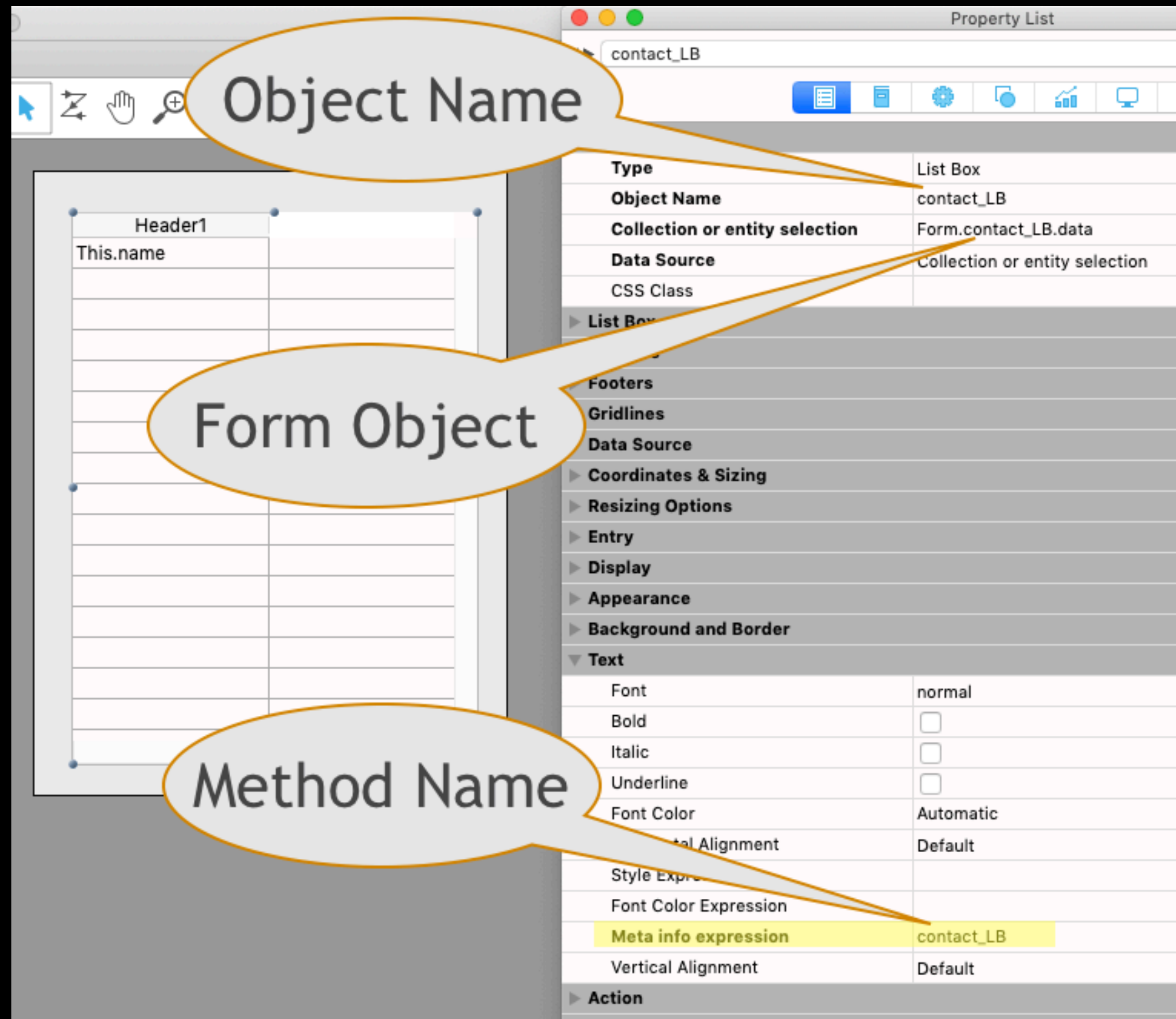
Type	List Box
Object Name	contact_LB
Collection or entity selection	Form.contact_LB.data
Data Source	Collection or entity selection
CSS Class	
List Box	
Footers	
Gridlines	
Data Source	
Coordinates & Sizing	
Resizing Options	
Entry	
Display	
Appearance	
Background and Border	
Text	
Font	normal
Bold	<input type="checkbox"/>
Italic	<input type="checkbox"/>
Underline	<input type="checkbox"/>
Font Color	Automatic
Horizontal Alignment	Default
Style Expression	
Font Color Expression	
Meta info expression	contact_LB
Vertical Alignment	Default
Action	

Because we use the same name doesn't mean it's the same thing

The screenshot displays a software development environment with a form object on the left and a 'Property List' panel on the right. The form object is a table with a header row labeled 'Header1' and a data row labeled 'This.name'. The 'Property List' panel shows various properties for the selected object, 'contact\_LB'. Three callouts are present: 'Object Name' points to the 'Object Name' property in the list; 'Form Object' points to the form object in the design view; 'Method Name' points to the 'Meta info expression' property, which has the value 'contact\_LB'.

Property	Value
Type	List Box
Object Name	contact_LB
Collection or entity selection	Form.contact_LB.data
Data Source	Collection or entity selection
CSS Class	
Text	
Font	normal
Bold	<input type="checkbox"/>
Italic	<input type="checkbox"/>
Underline	<input type="checkbox"/>
Font Color	Automatic
Font Color Expression	
Meta info expression	contact_LB
Vertical Alignment	Default

# Because we use the same name doesn't mean it's the same thing



- Using the same name makes it easy to identify what something is
- 4D is fine keep them separate (tokens!)
- It is important that *you* understand the differences

# To Review

We have established an approach to managing the key elements of a collection or entity based listbox. Including

# To Review

We have established an approach to managing the key elements of a collection or entity based listbox. Including

- A naming convention which defines a namespace for the listbox
- A single method as a list box controller
- A scheme for using the Meta Info Expression
- A scheme for identifying the contents and selected items of a listbox

# To Review - continued

# To Review - continued

- We can add data to the list box by simply setting the .data property to a collection or entity selection
- We can update or replace the Meta Info Expression
- The entire scheme is manage by a single method

But wait - there's more