

# Advanced REDCap Form Plugin

## API Documentation

April 10, 2012

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Package form.src</b>	<b>2</b>
1.1 Modules . . . . .	2
1.2 Variables . . . . .	2
<b>2 Module form.src.base_form</b>	<b>3</b>
2.1 Class FormCanvas . . . . .	3
2.1.1 Methods . . . . .	3
2.2 Class Form . . . . .	3
2.2.1 Methods . . . . .	3
<b>3 Module form.src.logic_parsers</b>	<b>7</b>
3.1 Variables . . . . .	7
3.2 Class LogicParser . . . . .	7
3.2.1 Methods . . . . .	7
3.2.2 Properties . . . . .	7
<b>4 Module form.src.print_form</b>	<b>8</b>
4.1 Functions . . . . .	8
4.2 Class PdfForm . . . . .	8
4.2.1 Methods . . . . .	8
4.2.2 Properties . . . . .	8
<b>5 Module form.src.redcap_form</b>	<b>10</b>
5.1 Class RedcapForm . . . . .	10
5.1.1 Methods . . . . .	10
<b>Index</b>	<b>17</b>

# 1 Package form.src

## 1.1 Modules

- **base\_form** (*Section 2, p. 3*)
- **logic\_parsers** (*Section 3, p. 7*)
- **print\_form** (*Section 4, p. 8*)
- **redcap\_form** (*Section 5, p. 10*)

## 1.2 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> None

## 2 Module `form.src.base_form`

### 2.1 Class `FormCanvas`

`reportlab.pdfgen.canvas.Canvas` —  
`form.src.base_form.FormCanvas`

#### 2.1.1 Methods

```
__init__(self, *args, **kwargs)
```

```
showPage(self)
```

```
save(self)
```

### 2.2 Class `Form`

#### 2.2.1 Methods

```
__init__(self, filename, font="Times-Roman", font_size=12)
```

```
set_font(self, font, font_size=12)
```

Change the font type and size.

Arguments: `font` – The string specifying the font to use when rendering text. Must be recognized by ReportLab in order to render correctly.

Keyword Arguments: `font_size` – Number specifying the size of the font to use when rendering text. Default is 12.

```
set_border(self, top, bottom, left, right)
```

Set the margins of the Form.

Arguments: `top` – Number specifying in inches the size of the top margin. `bottom` – Number specifying in inches the size of the bottom margin. `left` – Number specifying in inches the size of the left margin. `right` – Number specifying in inches the size of the right margin.

```
show_border(self)
```

Print the margin border on the current page.

```
render(self)
```

Render the PDF. Call only when finished constructing the form.

**add\_total\_page\_num**(*self*)

Add the (current page number)/(total page number) to right header.

**new\_page**(*self*, *break\_text=False*)

Go to the next page and set header/footer.

Arguments: *break\_text* – Boolean value; if set to True it add the continued on next page footer.

**new\_line**(*self*, *\*\*kwargs*)

Move the cursor to beginning of the next line.

**remove\_header\_footer**(*self*, *text*)

Remove the header/footer with the associated text. If text is not a current header/footer do nothing.

Arguments: *text* – The string to be removed from the header or footer.

**set\_header**(*self*, *text*, *location*, *font*, *all\_pg=False*)

Specify a header for the Document

Arguments: *text* – The string that will be set as the header. *location* – Specifies where the header should be placed. Can either be 'left', 'center' or 'right' *all\_pg* – Boolean value specifying whether the header should be on all pages.

**set\_footer**(*self*, *text*, *location*, *font*, *all\_pg=False*)

Set footer of the pdf.

Arguments: *text* – The string that will be set as the footer. *location* – Specifies where the footer should be placed. Can either be 'left', 'center' or 'right' *all\_pg* – Boolean value specifying whether the footer should be on all pages.

**print\_text**(*self*, *text*, *\*\*kwargs*)

Print the text to the current cursor(x,y) position, making sure text fits within the page borders and word wrap where necessary.

Arguments: *text* – The string to print to the form.

**form\_name**(*self*, *name*, *font\_name="Times-BoldItalic"*, *font\_size=20*)

Add form name to the form.

Arguments: *name* – The string that will be added as the form name.

Keyword Arguments: *font\_name* – The name of the font to render the form name. Make sure it is accepted by ReportLab in order to work properly. *font\_size* – Number specifying the size of the font to use to render the name.

**section\_name**(*self*, *name*, *font\_name*="Times-Bold", *font\_size*=14)

Add section name to the form.

Arguments: *name* – The string that will be added as the section name.

Keyword Arguments: *font\_name* – The name of the font to use to render the section name. Make sure it is accepted by ReportLab in order to work properly. *font\_size* – Number specifying the size of the font to use to render the name.

**start\_indent**(*self*)

Indent the start of the line. Will return if already in the center of the page to preserve legibility.

**stop\_indent**(*self*)

Unindent the start of the line. Will not go past the margin borders.

**date\_element**(*self*, *text*, *date\_format*='mmddyy')

Add date element to the form.

Arguments: *text* – The string to be rendered with the date boxes.

Keyword Arguments: *date\_format* – The string specifying the format for the date boxes. A space will be placed between different letters.

**time\_element**(*self*, *text*, *time\_format*='hhmm')

Add time element to the form.

Arguments: *text* – The string to be rendered with the time boxes. *time\_format* – The string specifying the format for the time boxes. A colon will be placed between different letters.

**text\_element**(*self*, *text*, *line\_len*=1)

Add a text element to the form.

Arguments: *text* – The text to render with the text line. *line\_len* – Number specifying the length of the blank line in inches.

**check\_box\_element**(*self*, *text*, *choices*)

Add a checkbox element to the form.

Arguments: *text* – The string that will render as the question text. *choices* – An array containing the choices as strings to be rendered as individual checkbox choices.

**radio\_element**(*self*, *text*, *choices*)

Add a radio button element to the form.

Arguments: *text* – The string that will render as the question text. *choices* – An array containing the choices as strings to rendered as individual radio button choices.

---

**slider\_element**(*self*, *text*, *choices*)

Add a slider element to the form.

Arguments: *text* – The string that will render as the question text. *choices* – String array containing 3 choices that will render along the top of the slider.

---

**note\_element**(*self*, *text*, *line\_nums*=8)

Add a note element to the form.

Arguments: *text* – The string that will render as the question text.

Keyword Arguments: *line\_nums* – Integer specifying the number of lines in the note box.

---

**SaveToFile**(*self*, *filename*, *pdfdata*)

Save the rendered pdf to a file.

Arguments: *filename* – The name of the file to save the pdf as. *pdfdata* – The pdfdata to write to the file.

### 3 Module *form.src.logic\_parsers*

#### 3.1 Variables

Name	Description
<code>--package--</code>	<b>Value:</b> <code>'form.src'</code>

#### 3.2 Class *LogicParser*

object  `form.src.logic_parsers.LogicParser`

##### 3.2.1 Methods

**`--init--(self)`**

`x.--init--(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.--init--` `exitit`(inherited documentation)

**`add_constraint(self, var, vals)`**

Adds a new constraint to the *LogicParser* instance.

Karguments: `var` – The variable name for the element. `vals` – A list of all the numeric acceptable values for the element.

**`evaluate(self, var_name, logic)`**

Return True if the element is not constrained by any elements in the constrained list, False otherwise.

Keyword arguments: `var_name` – The variable name for the element. `logic` – The string containing the branching logic for the element.

#### *Inherited from object*

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`, `--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

##### 3.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

## 4 Module `form.src.print_form`

### 4.1 Functions

**main**(*argv*=None)

### 4.2 Class PdfForm

object └─ **form.src.print\_form.PdfForm**

#### 4.2.1 Methods

**\_\_init\_\_**(*self*, *xml\_file*, *config\_file*=None)  
*x.\_\_init\_\_*(...) initializes *x*; see `help(type(x))` for signature  
 Overrides: `object.__init__` `exitit`(inherited documentation)

**revert\_indent\_val**(*self*)  
 Invert the indent value. The indent val is responsible for whether or not elements on the form are indented based on branching logic.

**add\_constraint\_list**(*self*, *const\_section*)  
 Add the list of section names that correspond to loaded the constraint file. These sections will be used to apply the appropriate constraints.  
 Arguments: *const\_section* – The section(s) in the *config\_file* to use along with the base section

**process**(*self*, *const*)  
 Parse and create the RedcapForm associated with the REDcap XML data dictionary.

#### *Inherited from object*

`__delattr__`(), `__format__`(), `__getattr__`(), `__hash__`(), `__new__`(), `__reduce__`(), `__reduce_ex__`(), `__repr__`(), `__setattr__`(), `__sizeof__`(), `__str__`(), `__subclasshook__`()

#### 4.2.2 Properties



Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

## 5 Module `form.src.redcap_form`

### 5.1 Class `RedcapForm`



#### 5.1.1 Methods

<b><code>__init__(self, *args, **kwargs)</code></b> Overrides: <code>form.src.base_form.Form.__init__</code>
<b><code>setup(self)</code></b> Setup initial parameters for a REDCap form
<b><code>add_to_header_box(self, element, callback, font, choices=None)</code></b>
<b><code>print_header_box(self)</code></b> Print the Header box on the top right of every page.
<b><code>no_print(self, question)</code></b> Dummy method for questions that should not be rendered.
<b><code>new_page(self, break_text=False)</code></b> Set the cursor to the top of the next page. Overrides: <code>form.src.base_form.Form.new_page</code>
<b><code>render(self)</code></b> Render the form when done adding elements. Overrides: <code>form.src.base_form.Form.render</code>
<b><code>integer_element(self, question)</code></b> Add an integer element to the form. Arguments: <code>question</code> – The question text to be rendered.

**number\_element**(*self*, *question*)

Add a number element to the form.

Arguments: question – The question text to be rendered.

**text\_element**(*self*, *question*)

Add a text element to the form.

Arguments: question – The question text to be rendered.

Overrides: form.src.base\_form.Form.text\_element

**date\_ymd**(*self*, *question*)

Add a date element to the form with the format yyyy/mm/dd.

Arguments: question – The question text to be rendered.

**date\_dmy**(*self*, *question*)

Add a date element to the form with the format dd/mm/yyyy.

Arguments: question – The question text to be rendered.

**time**(*self*, *question*)

Add a time element to the form with the format hh:mm.

Arguments: question – The question text to be rendered.

**time\_mm\_ss**(*self*, *question*)

Add a time element to the form with the format mm:ss.

Arguments: question – The question text to be rendered.

**datetime\_mdy**(*self*, *question*)

Add a datetime element to the form with the format mm/dd/yyyy hh:mm.

Arguments: question – The question text to be rendered.

**datetime\_dmy**(*self*, *question*)

Add a datetime element to the form with the format dd/mm/yyyy hh:mm.

Arguments: question – The question text to be rendered.

**datetime\_ymd**(*self*, *question*)

Add a date element to the form with the format yyyy/mm/dd hh:mm.

Arguments: question – The question text to be rendered.

**datetime\_sec\_dmy**(*self*, *question*)

Add a date element to the form with the format dd/mm/yyyy hh:mm:ss.

Arguments: question – The question text to be rendered.

**datetime\_sec\_mdy**(*self*, *question*)

Add a date element to the form with the format mm/dd/yyyy hh:mm:ss.

Arguments: question – The question text to be rendered.

**datetime\_sec\_ymd**(*self*, *question*)

Add a date element to the form with the format yyyy/mm/dd hh:mm:ss.

Arguments: question – The question text to be rendered.

**descriptive\_element**(*self*, *text*)

Add a descriptive element to the form.

Arguments: question – The text to be rendered.

**truefalse\_element**(*self*, *text*)

Add a truefalse element to the form with the format yyyy/mm/dd.

Arguments: question – The question text to be rendered.

**yesno\_element**(*self*, *text*)

Add a yesno element to the form with the format yyyy/mm/dd.

Arguments: question – The question text to be rendered.

**dropdown\_element**(*self*, *text*, *choices*)

Add a dropdown element to the form with the format yyyy/mm/dd.

Arguments: question – The question text to be rendered.

**note\_element**(*self*, *text*)

Add a note element to the form with the format yyyy/mm/dd.

Arguments: question – The question text to be rendered.

Overrides: form.src.base\_form.Form.note\_element

**sql\_element**(*self*, *text*)

A Dummy sql element method. Does not add anything to the form.

Arguments: question – The question text.

**SaveToFile**(*self*, *filename*, *pdfdata*)

Save the rendered pdf to a file.

Arguments: filename – The name of the file to save the pdf as. pdfdata – The pdfdata to write to the file.

**add\_total\_page\_num**(*self*)

Add the (current page number)/(total page number) to right header.

**check\_box\_element**(*self*, *text*, *choices*)

Add a checkbox element to the form.

Arguments: text – The string that will render as the question text. choices – An array containing the choices as strings to be rendered as individual checkbox choices.

**date\_element**(*self*, *text*, *date\_format*='mmddyy')

Add date element to the form.

Arguments: text – The string to be rendered with the date boxes.

Keyword Arguments: date\_format – The string specifying the format for the date boxes. A space will be placed between different letters.

---

**form\_name**(*self*, *name*, *font\_name*="Times-BoldItalic", *font\_size*=20)

---

Add form name to the form.

Arguments: *name* – The string that will be added as the form name.

Keyword Arguments: *font\_name* – The name of the font to render the form name. Make sure it is accepted by ReportLab in order to work properly.  
*font\_size* – Number specifying the size of the font to use to render the name.

---

**new\_line**(*self*, *\*\*kwargs*)

---

Move the cursor to beginning of the next line.

---

**print\_text**(*self*, *text*, *\*\*kwargs*)

---

Print the text to the current cursor(x,y) position, making sure text fits within the page borders and word wrap where necessary.

Arguments: *text* – The string to print to the form.

---

**radio\_element**(*self*, *text*, *choices*)

---

Add a radio button element to the form.

Arguments: *text* – The string that will render as the question text. *choices* – An array containing the choices as strings to rendered as individual radio button choices.

---

**remove\_header\_footer**(*self*, *text*)

---

Remove the header/footer with the associated text. If text is not a current header/footer do nothing.

Arguments: *text* – The string to be removed from the header or footer.

---

**section\_name**(*self*, *name*, *font\_name*="Times-Bold", *font\_size*=14)

---

Add section name to the form.

Arguments: *name* – The string that will be added as the section name.

Keyword Arguments: *font\_name* – The name of the font to use to render the section name. Make sure it is accepted by ReportLab in order to work properly. *font\_size* – Number specifying the size of the font to use to render the name.

---

**set\_border**(*self, top, bottom, left, right*)

Set the margins of the Form.

Arguments: top – Number specifying in inches the size of the top margin.  
bottom – Number specifying in inches the size of the bottom margin. left –  
Number specifying in inches the size of the left margin. right – Number  
specifying in inches the size of the right margin.

---

**set\_font**(*self, font, font\_size=12*)

Change the font type and size.

Arguments: font – The string specifying the font to use when rendering text.  
Must be recognized by ReportLab in order to render correctly.

Keyword Arguments: font\_size – Number specifying the size of the font to use  
when rendering text. Default is 12.

---

**set\_footer**(*self, text, location, font, all\_pg=False*)

Set footer of the pdf.

Arguments: text – The string that will be set as the footer. location – Specifies  
where the footer should be placed. Can either be 'left', 'center' or 'right' all\_pg  
– Boolean value specifying whether the footer should be on all pages.

---

**set\_header**(*self, text, location, font, all\_pg=False*)

Specify a header for the Document

Arguments: text – The string that will be set as the header. location –  
Specifies where the header should be placed. Can either be 'left', 'center' or  
'right' all\_pg – Boolean value specifying whether the header should be on all  
pages.

---

**show\_border**(*self*)

Print the margin border on the current page.

---

**slider\_element**(*self, text, choices*)

Add a slider element to the form.

Arguments: text – The string that will render as the question text. choices –  
String array containing 3 choices that will render along the top of the slider.

---

**start\_indent**(*self*)

Indent the start of the line. Will return if already in the center of the page to preserve legibility.

---

**stop\_indent**(*self*)

Unindent the start of the line. Will not go past the margin borders.

---

**time\_element**(*self*, *text*, *time\_format*='hhmm')

Add time element to the form.

Arguments: *text* – The string to be rendered with the time boxes. *time\_format* – The string specifying the format for the time boxes. A colon will be placed between different letters.



## Index

- form (*package*)
  - form.src (*package*), 2
    - form.src.base\_form (*module*), 3–6
    - form.src.logic\_parsers (*module*), 7
    - form.src.print\_form (*module*), 8–9
    - form.src.redcap\_form (*module*), 10–16