



Table of Contents

Overview	2
What is a PDF rendered visualforce page?	2
Configuring the PDFer button	2
What is a Button Configuration?	2
Creating a new Button Configuration	3
Adding PDFer button to an Object Page Layout	5
Adding PDFer button to an Object Listview	5
Using from a Salesforce Object	7
Using from a Salesforce Listview	7
After the PDFer button is clicked	7
Setting up to use with another Standard or Custom Object	8
Create a new Visualforce page	8
Create a new Button	9
Add the Button to the Object's Layout	9
Optional Attributes	9
Permission Sets	10
PDFer User	10
PDFer Admin	10
Use Cases	10
Emailing and saving a pdf then recording it has been done	10



Overview

Create, Save and Email PDFs & Files from Salesforce records and listviews.

The PDFer button can be added to any Standard or Custom Salesforce Object page layout and any listview (if the Object in question supports them).

Using the App, a Button can be created and configured to appear when the PDFer button is clicked.

Clicking the button can then save and email one or more pdf rendered visualforce pages (plus other Files) using the selected Salesforce records and automatically record the process is complete for each record.

What is a PDF rendered visualforce page?

A visualforce page can be set up in Salesforce to render as a PDF by changing the `<apex:page>` tag to include the 'RenderAs' attribute.

```
<apex:page standardController="contact" RenderAs="pdf">
  <b>{!contact.Id}</b>
  <i>{!contact.name}</i>
</apex:page>
```

The above is a very simple example of a visualforce page that will render as a pdf, and contains details of a Contact. HTML tagging can be used to format the content as required. So very rich formatting is possible.

This example could be used by PDFer from a Button configured for the Contact Object.

Read more at:-

https://developer.salesforce.com/docs/atlas.enus.pages.meta/pages/pages_output_pdf_renderas.htm

Configuring the PDFer button

The PDFer button can be added to the detail page or listview of any Account, Case, Contact, Lead or Opportunity. It can also be configured and used with any other Standard or Custom Object.

When the PDFer button is clicked any Buttons configured for the Object will appear for the User to select.

What is a Button Configuration?

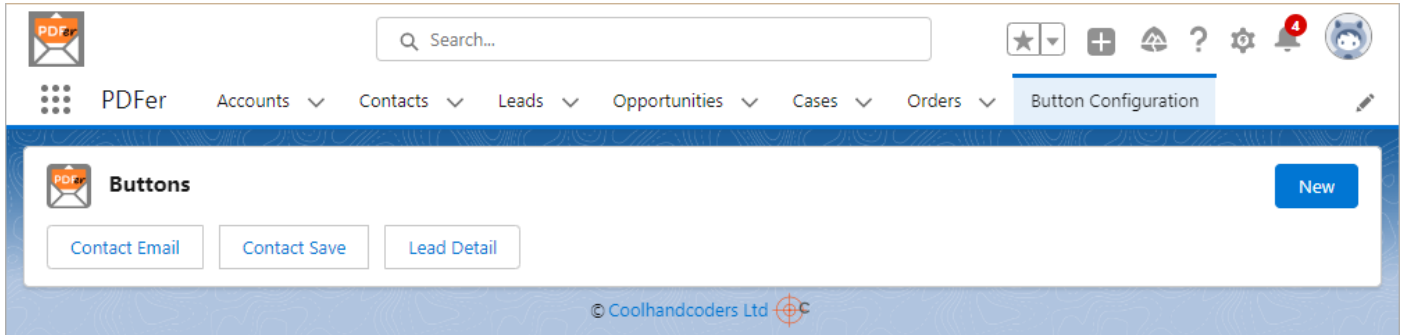
A Button Configuration stores information for a Button that should appear when the PDFer button is clicked.

This information includes the label for the button, the Object name whose detail page and listview it should be appear on, one or more visualforce pages to be rendered as a pdf (and the filenames to give each), one or more other Files to be included, whether the rendered pdfs & Files should be saved, emailed (or both), and if a Salesforce Object should be flagged after processing.



Creating a new Button Configuration

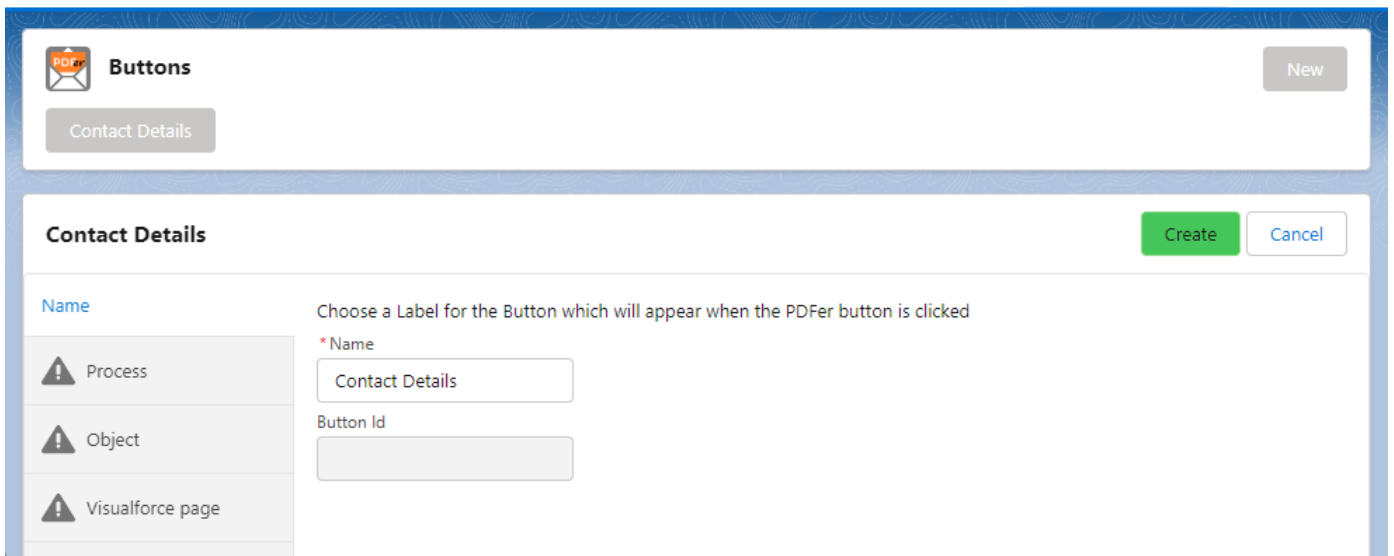
Click on the [Button Configuration] Tab of the App



Click on the [New] button



Then enter a [Name] for the Button which will be used as the label for the button





Choose whether the button should save, email or do both

- If the Files and pdf should be saved in Salesforce, then select [Save] and enter the required details in each of the vertical Tabs
- If the Files and pdf should be emailed, then select [Email] and enter the required details in each of the vertical Tabs
- If the Files and pdf should be emailed **and** saved in Salesforce, then select [Email & Save] and enter the required details in each of the vertical Tabs

The screenshot shows the 'Buttons' configuration page in Salesforce. The 'Contact Details' tab is selected. The 'Process' dropdown is set to 'Save'. The 'Name' field is empty. The 'Object' and 'Visualforce page' fields are also empty. The 'Create' and 'Cancel' buttons are visible at the top right of the configuration area.

Once all the required details are filled in, click the [Create] Button to save the Button Configuration.

The screenshot shows the 'Buttons' configuration page in Salesforce. The 'Contact Details' tab is selected. The 'Name' field is filled with 'Contact Details'. The 'Process' dropdown is set to 'Save'. The 'Object' and 'Visualforce page' fields are also empty. The 'Create' and 'Cancel' buttons are visible at the top right of the configuration area.



It will then appear as a button at the top of the screen where it can be selected and details updated.

Adding PDFer button to an Object Page Layout

Navigate to the Page layout of the required Object, select [Mobile & Lightning Actions], then drag the [PDFer] button to the [Salesforce Mobile and Lightning Experience Actions] section, then click [Save].

Adding PDFer button to an Object Listview

Navigate to the required Object Listview, then select 'PDFer' from the [Available Buttons] list, click the [Add] button to add it to the [Selected Buttons] list, then click [Save].





Search Setup

SetupHomeObject Manager

SETUP > OBJECT MANAGER

Contact

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Hierarchy Columns

Triggers

Flow Triggers

Validation Rules

Contacts List View

Customize the buttons on the Contacts list view:

- To remove any standard buttons, remove the check next to the standard button name.
- To add custom buttons, select them and click Add.

Standard Buttons

☒ Open in Quip [OpenListInQuip]
☒ New From Document [NewFromDocument]
☒ New [NewContact]
☒ Add to Campaign [AddToCampaign]
☒ Import [Import]
☒ Add to Campaign [MassAddToCampaign]
☒ Add to Call List [MassCreateCallList]
☒ Send List Email [SendListEmailAction]
☒ Send Message [SendBulkMessageAction]
☒ Printable View [PrintableListView]

Custom Buttons

Available Buttons

PDFer

Add

Remove

Selected Buttons

--None--

Up

Down

List View Actions in Lightning Experience

[New Quick Action](#)

SaveCancel



Using from a Salesforce Object

Navigate to the detail page of the required Salesforce record, then click the [PDFer] button

The screenshot shows the Salesforce interface for a Contact record. The top navigation bar includes the PDFer logo, a search bar, and various utility icons. The main navigation menu has tabs for Accounts, Contacts, Leads, Opportunities, Cases, Orders, and Button Configuration. The 'Contacts' tab is selected, showing the detail page for 'Mr. Sean Forbes'. The record header includes buttons for 'PDFer', 'Printable View', and 'Change Owner'. Below the header, the record details are displayed: Title (CFO), Account Name (Edge Communications), Phone (512) 757-6000, Email (sean@edge.com), and Contact Owner (Rhydian Coombs).

Using from a Salesforce Listview

Select the records in the required listview, then click the [PDFer] button

The screenshot shows the Salesforce interface for a Contacts listview. The top navigation bar includes the PDFer logo, a search bar, and various utility icons. The main navigation menu has tabs for Accounts, Contacts, Leads, Opportunities, Cases, Orders, and Button Configuration. The 'Contacts' tab is selected, showing the 'Recently Viewed' listview. The listview header includes buttons for 'New' and 'PDFer'. Below the header, the listview content is displayed, showing a table of contacts. Three records are selected (checked): Sean Forbes, Rose Gonzalez, and Josh Davis. The table columns include Name, Account Name, Phone, Email, and Contact Owner.

After the PDFer button is clicked

If only one Button is configured for a Salesforce Object then the screen will appear as below...

The screenshot shows the Salesforce interface after clicking the PDFer button. A modal titled 'Contact PDFer' is displayed. The modal header includes the PDFer logo and a button for 'Contact Details'. The main content area shows '3 record(s) selected' and a button for 'Test Contact Template'. Below this, there is a green button for 'Email and Flag (3)' and a blue button for 'Close'. At the bottom of the modal, there is a 'Preview' section with a blue button. The footer of the modal displays the copyright notice '© Coolhandcoders Ltd'.



If multiple Buttons are configured for a Salesforce Object then a Button first needs to be selected...



Setting up to use with another Standard or Custom Object

There are 3 steps required...

Create a new Visualforce page

Create a new Visualforce page replacing the standardController attribute with the API Name of the required Salesforce Object.

Paste the following for a button that will appear on the page layout of the Salesforce Object:-

```
<apex:page standardController="CustomObject__c" extensions="pdfer.chc_CTL_PDFer_Process">
<apex:includeLightning />
<div id="LightningAppContainerId"></div>
<script>
    $Lightning.use("pdfer:chc_LA_PDFer_Process", function() {
        $Lightning.createComponent("pdfer:chc_LC_PDFer_Process",
            {
                "atObjectLabel": "{!pObjectLabel}",
                "atObjectName": "{!pObjectName}",
                "atRecordcount": "{!pRecordCount}",
                "atObjectIcon": "{!pObjectIcon}",
                "atClosePageURL": "{!pClosePageURL}",
                "atRecordIdList": "{!pRecordIdList}"
            },
            "LightningAppContainerId",
            function(component) {});
    });
</script>
</apex:page>
```

Paste the following for a button that will appear on the Salesforce Object's Listviews screen:-

```
<apex:page standardController="CustomObject__c" recordSetVar="records" extensions="pdfer.chc_CTL_PDFer_Process">
<apex:includeLightning />
<div id="LightningAppContainerId"></div>
<script>
    $Lightning.use("pdfer:chc_LA_PDFer_Process", function() {
        $Lightning.createComponent("pdfer:chc_LC_PDFer_Process",
            {
                "atObjectLabel": "{!pObjectLabel}",
                "atObjectName": "{!pObjectName}",
                "atRecordcount": "{!pRecordCount}",
                "atObjectIcon": "{!pObjectIcon}",
                "atClosePageURL": "{!pClosePageURL}",
                "atRecordIdList": "{!pRecordIdList}"
            },
            "LightningAppContainerId",
            function(component) {});
    });
</script>
</apex:page>
```



Create a new Button

Create a new Button for the required Salesforce Object labelled 'PDFer' (or whatever label you want) and set its Content to the new Visualforce page.

Add the Button to the Object's Layout

Add the Button to the Object's Page Layout or Listview Button Layout. The Button will then work in the same way as the built in PDFer Buttons provided.

Optional Attributes

There are some optional attributes that can be included in the Visualforce markup.

atButtonId: Use this when there are multiple Button Configurations set up for the same Salesforce Object and you want your button to open for only one of the Button Configurations.

The Button Id value to pass for this attribute can be found in the Name tab of the required Button Configuration in the PDFer App.

e.g. "atButtonId": "a1eR5000001vPsWXYZ"

atEmailTemplateFolderOverride: Use this when you want to override the Folder defined in the Email Template tab of the required Button Configuration in the PDFer App.

Logic can then be used to decide which Folder the User can select an Email Template from 'On the fly'.

e.g. For a button whose standardcontroller is the Account Object, you could pass the following logic:-

"atEmailTemplateFolderOverride": "{!if(Account.Group__c == 'Global UK','UK_Folder','NonUK_Folder')}}"

The above will check the Account's (or first Account if from a Listview) [Group Company] value and use this to decide which Folder the User can select an Email Template from.

Following is an example of the markup with the two optional attributes included:-

```
<apex:page standardController="Account" extensions="pdfer:chc_CTL_PDFer_Process">
  <apex:includeLightning />
  <div id="LightningAppContainerId"></div>
  <script>
    $Lightning.use("pdfer:chc_LA_PDFer_Process", function() {
      $Lightning.createComponent("pdfer:chc_LC_PDFer_Process",
        {
          "atObjectLabel": "{!pObjectLabel}",
          "atObjectName": "{!pObjectName}",
          "atRecordCount": "{!pRecordCount}",
          "atObjectIcon": "{!pObjectIcon}",
          "atClosePageURL": "{!pClosePageURL}",
          "atRecordIdList": "{!pRecordIdList}",
          "atButtonId": "a1eR5000001vPsWXYZ",
          "atEmailTemplateFolderOverride": "{!if(Account.Group__c == 'Global UK','UK_Folder','NonUK_Folder')}}"
        },
        "LightningAppContainerId",
        function(component) {});
    });
  </script>
</apex:page>
```



Permission Sets

There are two Permission *PDFer User* & *PDFer Admin* that provide access to functionality of the App.

PDFer User

Enables a User to use the PDFer Button on the Account, Contact, Lead, Opportunity & Case Page Layouts & Listviews.

PDFer Admin

Enables a User to use the PDFer Button on the Account, Contact, Lead, Opportunity & Case Page Layouts & Listviews & create, update & delete the Button Configurations via the Button Configuration Tab.

Use Cases

Emailing and saving a pdf then recording it has been done

"I want to email Contacts their details as a pdf, then attach the pdf to the contact's account record and record the date emailed in the contact record"

Create a visualforce page

Create your visualforce page detailing the contact's details. For example...

```
<apex:page standardController="contact" RenderAs="pdf">
  Id: {!contact.Id}
  Name: {!contact.name}
  Account: {!contact.Account.name}
</apex:page>
```

Save the visualforce page with the name 'Contact Details'

Create an Email Template

Create an Email Template that includes the required text and mail merge fields. For example...

Add a classic email template with an [Email Template Name] of 'Contact Details Template', [Subject] 'Here are your Contact Details' and [Email Body] of

Dear {!Contact.Name}

The attached pdf contains your Contact details in Salesforce.

Your Account is {!Account.Name}

Regards.

Tick [Available for use] then click [Save]

Configure the button to do this

Open the PDFer App

Click the [Button Configuration] Tab

Click the [New] button

Enter the label for the button. For example 'Contact Details'



Click the [Process] tab and select 'Email & Save'

Click the [Object] tab and select 'Contact'

Click the [Visualforce page] tab and select the 'Contact Details' page created earlier

Click the [Filename] tab and enter the name for the pdf generated. For example '[Name]_ContactDetails.pdf'

Click the [Email] tab and choose the 'Contact ID' from the 'To' list. Choose '(Account) Account Id' from the 'Related Object' list

Click the [Email Template] tab and choose the [Folder] of containing the 'Contact Details Template' Email Template.

Click the [Save To] tab and select 'Account (Account ID)'

Click the [Flag] tab, select 'Contact ID' as the Object Id and the choose a date Field in the 'Flag Field' list

[Add the PDFer button to the Contact Listview](#)

See [Adding PDFer button to an Object Listview](#) sections