SIMPLE SIGNATURE

SALESSFORCE CLASSIC CONFIGURATION

ToAFinish!

SIMPLE SIGNATURE FOR SALESFORCE

Salesforce Classic Configuration

12407 MoPac Expwy N • Suite 250
Austin, TX 78758

www.SimpleSignatureApp.com

Table of Contents

Version 2021.05.31

Chapter 1

Standard Configuration in Classic

Introducing all the key components of the Simple Signature App for Salesforce.com

This manual will help you through the implementation of the Simple Signature app for your Salesforce.com Org. You can read this chapter if you desire to do so, but most users can probably start at the next chapter, where the installation and configuration instructions begin.

Launching from the Parent Object

Instructions to set up a "Sign" button on the Contact record as an example, are below. This button can be used in Lightning and in Classic. It can be used in Mobile somewhat, but for various reasons we recommend going to the Lightning version(s) in the next section.

1) Still in the Lightning, desktop version of Salesforce, with Admin permissions, at the top of the page click on Setup → Object Manager and click on **Signature**.

2) In the menu on the left, click on the **Fields & Relationship** section and then press the **New** button to create a Lookup field to the object you want to have as the Signature parent. Contact, Lead and Opportunity are already done for you.

   ** Note: Do not duplicate lookup fields in the Signature object. If you need to change the name of a lookup, use the translation feature, rather than creating a new field.*

3) Again at the top-right of the page click on Setup → Object Manager and then click on the **Parent Object**.

4) Now click on the **Buttons, Links and Actions** section, and click on the **New Button or Link** button.

5) Enter "Sign" for the Label and select **Detail Page Button** as the Display Type.

6) For the Behavior you can select "Display in existing window without sidebar or header, and for the Content Source, choose **URL**.

7) Now, for the actual URL, you must customize it depending on the parent object that you selected. If you are using Contacts, then the URL would be this:

```
/apex/signature__Signature
?parentid={!Contact.Id}
&backtoparent=1
```

Parameter	Description
parentid	This will link the new Signature with a parent record of any type. For example, for the Contact object, you would use the standard **{!Contact.Id}** formula field notation.
backtoparent	1 to take the user back to the parent record after the signature process completes, 0 (default) to leave the user on the newly created Signature record.

8) Save the Button. Skip to page 18 for adding the button to the layout.

Launching from the Signature Related List

Instructions to set up a "New Signature" button on the Signatures related list for the Contact as an example, are below:

1) In the "Lightning" desktop version of Salesforce, with Admin permissions.

2) At the top of the page click on Setup → Object Manager, then click **Signature**.

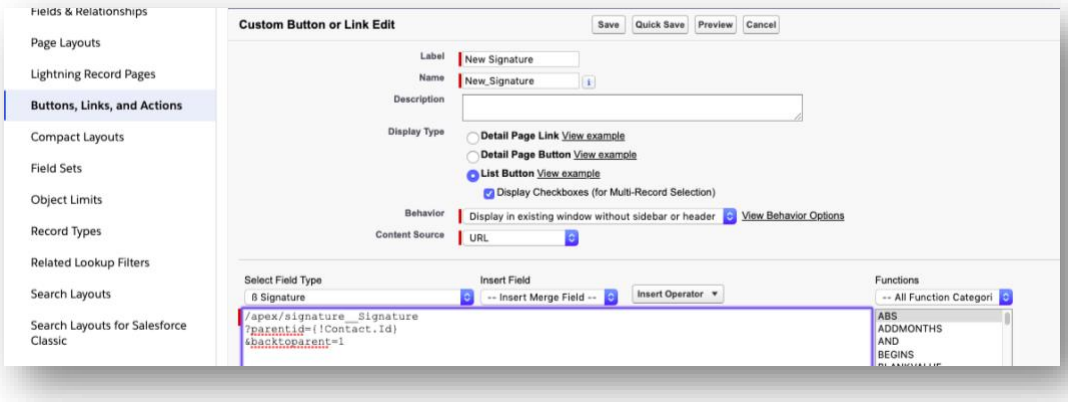
3) In the menu on the left, click on the **Fields & Relationship** section and then press the **New** button to create a Lookup field to the object you want to have as the Signature parent. Contact, Lead and Opportunity are already done for you.

   ** Note: Do not duplicate lookup fields in the Signature object.*

4) Now click on the **Buttons, Links and Actions** section, and click on the **New Button or Link** button.

5) Enter "New Signature" for the Label and then select **List Button** as the Display Type (and unselect the Display Checkboxes option).

6) For the Behavior you can select "Display in existing window without sidebar or header, and for the Content Source, choose **URL**. Something like this:



7) Now, for the actual URL, you must customize it depending on the parent object that you selected. If you are using Contacts, then the URL would be this:

```
/apex/signature__Signature
?parentid={!Contact.Id}
&backtoparent=1
```

8) Save the Button. Instructions on adding it to the layout are on the next page.

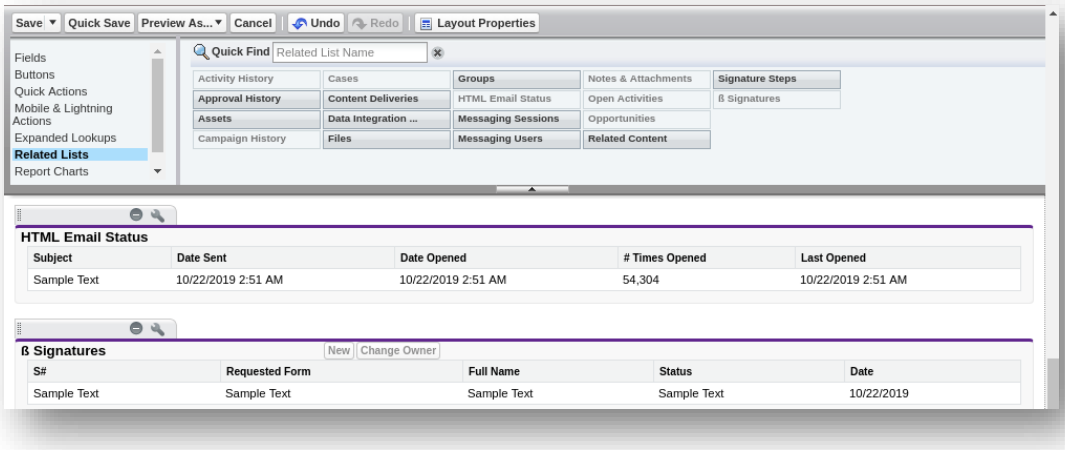
Adding the Signature Button to a Layout

Instructions to set up a "New Signature" button on the Signatures related list for the parent object, as described on the prior page. In this case, we will use Contacts as the example:

ADDING A BUTTON ON THE PARENT LAYOUT

1) Go back to Setup → Object Manager and click on **Contacts** and then **Page Layouts** from the left menu. (Note: if you selected a different object as the parent, then go to the page layouts for that object instead.)

2) Select "Buttons". Drag the button into the appropriate area.



3) We do not recommend using Classic buttons in Lightning, though they can be dragged into the Lightning area from the "Mobile & Lightning Actions" area. We recommend to instead use Actions intended for Lighting and Mobile as described in the standard Implementation Guide. Save the Layout and test it out.

ADDING A BUTTON ON THE SIGNATURES RELATED LIST

1) Go back to Setup → Object Manager and click on **Contacts** and then **Page Layouts** from the left menu. (Note: if you selected a different object as the parent, then go to the page layouts for that object instead.)

2) Go to "Related Lists" and drag down the Signatures related list onto the layout.

3) Edit the Signature related list to remove the standard "New" button (and any other you don't want) and add the new button you just created. Save the Layout and test it out.

You are done! Happy Signing!

Chapter 2

Advanced Configuration (Paid)

The steps necessary to set up paid functionality in Classic

The initial steps in the configuration are completed by first installing the package from the AppExchange.com app store. Make sure to **Install for All Users**. Even though only an Admin will configure it, you need to install for All Users so that they can access the pages correctly.

Advanced Parameters

The **Paid Version** of Simple Signature supports several other parameters that can be added to the URL when using the Visualforce option in order to achieve advanced functionality. (If you are using Lightning or Lightning Web components, you do not need to worry about the URL, as parameters are passed via the Signature Experience Id.).

For example, if you want to have an email automatically sent out requesting a signature from the contact attached to the Signature record, want to change what the form looks like, and want to attach the final PDF on the Contact, then you would construct a URL like this:

```
/apex/signature__Signature
?parentid={!Contact.Id}
&backtoparent=1
&pdftemplate=c__MyVFPage
&emailpdf=1
&attachpdftoparent=1
&attachpdf=0
&requestedPDFName={!Contact.Name}_SignOff2020
&defaultvalue_Phone__c={!Contact.MobilePhone}
&defaultvalue_Email__c={!Contact.Email}
&contactField=Customer_Contact__c
&requestedform=My Form Name
```

As of recent versions of the app, you are also able to use Experience IDs in Classic, so if an Experience is created in the Configure page, the above URL could be shortened to:

```
/apex/signature__Signature
?parentid={!Contact.Id}
&experienced=XXXXXXXXXXXX
```

A list of supported parameters in the paid version (for Classic) is shown below:

Parameter	Description	Type
allowrequest	1 to include a "Signature Request" button on the Signature Form, which when used, creates the Signature record with a "Requested" status, 2 is the same as 1 but it does not include the Signature pad at all - for when it always must be a signature request , 0 (default) to do nothing.	Requests
attachpdf	1 to attach the PDF generated to the Signature record, 0 (default) to do nothing.	Universal
attachpdftoparent	1 to attach the PDF generated to the Parent record, 0 (default) to do nothing.	Universal
backtoparent	1 to go back to the parent record when done signing , 0 (default) will send you to the Signature record created.	Universal
communityname	Name of the Community, to be used in URL handling. Use only when the Signature experience is to be used in a Salesforce Community.	Universal
contactField	If your parent object is not the Contact record, but you still want to attach a Contact to the Signature record (you must to this if you want any of the PDF/email automation), then you need to tell Simple Signature where the Contact field is supposed to come from. For this, simply put the API name (not the label) of the Contact lookup field in the parent object.	Universal
defaultfilterfield_	Please contact technical support.	Advanced
defaultfilterlookupvalue_	Please contact technical support.	Advanced
defaultfiltervalue_	Please contact technical support.	Advanced
defaultvalue_XXX	The "defaultvalue_" parameter is dynamic, the parameter name always starts the same, but then must use a name of a formula field in the Signature object. For example: defaultvalue_InspectionDate__c or defaultvalue_CustomerContact__c. More on using this to achieve read-only values on the Signature form in the next section.	Universal
deletesigantureafter	1 to delete the PNG image of the signature after the final PDF is generated , 0 (default) to do nothing.	Universal
emailpdf	1 to email the PDF generated to the contact attached, 0 (default) to do nothing.	Universal
embedpdf	1 to show the PDF to be signed in an iFrame above the signature form at the time of signing , 0 (default) to do nothing.	Universal
fieldheader	API name of the field(s) to show as Headers on the signature form. Separate multiple fields with commas.	Universal

Parameter	Description	Type
fieldhidelabel	API name of the field(s) to show with no label on the signature form. Separate multiple fields with commas.	Universal
fieldreadonly	API name of the field(s) to make read-only on the signature form. Separate multiple fields with commas.	Universal
fieldset	FieldSet to use on the Signature Form. Uses SignatureFields as the default if not included.	Universal
hideblanks	1 to hide the input fields if they are blank, 0 (default) to show all fields on the signature form.	Universal
hidecancel	1 to hide the Cancel button on the Signature Form , 0 (default) to do nothing.	Universal
language	Language of the form.	Universal
lookupdisplayfields_	Please contact technical support.	Advanced
lookupreturnfield_	Please contact technical support.	Advanced
multistep	1 to create a Step record when the Signature request is created, 0 (default) to remain a single Signature request.	Requests
nosignature	1 to allow submitting the form without signing, 0 (default) to do nothing.	Requests
optionimagename	Please contact technical support.	Advanced
optionpopulatedfrom	Please contact technical support.	Advanced
optionsavedin	Please contact technical support.	Advanced
parameterfield	To be used only in URL-based Visualforce pages. The API name of the field which contains parameters.	Deprecated
parentid	The Id of the record to be considered the parent of the Signature.	Universal
pdftemplate	The name of the Visualforce page to be used as the PDF template that is populated and then emailed to the contact attached to the Signature record once the submit button is pressed. Note: The name must be pre-pended with "c__".	Universal
requestedform	The name of the form that is being requested; used for differentiating Signature requests.	Universal
requestedpdfname	The name that you want the eventual PDF attachment to have. You can hard-code as well as include fields from the value. For example, for Contact parent object, where the Last Name was "Smith", if you used the parameter this way: "requestedPDFName={!Contact.Name}_Sign2020", then the resulting PDF would be called: "Smith_Sign2020.pdf".	Universal
returl	URL signer is to be sent to upon completing the Signature Form.	Universal
rurl	URL signer is to be sent to upon completing the Signature Form.	Universal
sdocstype	This is useful for s-Docs integration only. Do not use this, if you do not have s-Docs installed. The value of this parameter should be the API name of the parent object. For example: Contact. With this parameter, after the signature process, you will be re-directed to select an s-Docs template.	Advanced
showattachmentid	1 to show the PDF in the signautre form if the AttachmentID field has an ID populated, 0 (default) to do nothing.	Advanced
twosignatures	1 to display two separate signature pads on the same signature form, 0 (default) to show only 1 signature pad.	Universal

** Note: the last few parameters are complicated, and may need some help from our support team. Feel free to reach out to support@toafinish.com with questions.*

Creating a Custom Signature Page

Most companies will be able to use the default signature page that comes with the app, but if you find yourself in need to changing more than just a couple fields, or having advanced functionality on this page, you can create your own version of the page that starts right where the standard one ends, and allows you to build onto it as needed.

* Note: A fairly advanced user will be able to follow these instructions to replicate the standard functionality using a custom page, but adding new functionality will require a user with development experience. Contact support@toafinish.com if you need help in this task. We can answer questions, and if you need us to develop the page, we offer those services as well. Let us know.

1) First, you will need to create a new Visualforce page. Go to Setup → Custom Code → Visualforce Pages and click the "New" button. If the "New" button is not available, you will need to speak with your Administrator to get additional rights.

2) Once you see a new bank Visualforce page, fill out the Label and Name fields and then delete all of the default content that is in the main Visualforce Markup area. Now you are ready to put the actual code in.

GitHub.com Simple Signature Sample Code Repository
https://github.com/ToAFinish/simplesignature

3) Go to the GitHub repository above and go into the "pages" folder and click on the "CustomSignature_Lightning" entry. Copy all of the text in this page (without the line numbers), starting with <apex:page and ending with </apex:page>. Paste all of this text into the Visualforce Markup area in Salesforce and then Save the page.

Congratulations. You've done the hardest part. Now - you need to make sure everybody can see your page.

4) First, go back to Setup > Custom Code > Visualforce Pages and find your new page on the list. Click on the "Security" link next to the name of the new page. A list of the profiles that can view this page will show up, you want to select all of them, or at least all of the ones who will use Simple Signature, and give them access. Save.

5) We recommend optionally giving your new page a Tab style. To do so, go to Setup > User Interface > Tabs and press "New" on the **Visualforce Tabs** section. Choose the page that you created, name it, and then select a color and icon that you like. Save.

> *Note: You might need to change the Salesforce mobile tab that you are using to this new tab, as described on page 9.*

6) Now you need to go back and modify any custom URLs you have created using the instructions starting on page 13. Change the "signature__Signature" page name to the name of your new page. For example, if the URL looks like this:

```
/apex/signature__Signature?parentid={!Contact.Id}&backtoparent=1
```

And your new page is called "CustomSignature", then the new URL would be:

```
/apex/CustomSignature?parentid={!Contact.Id}&backtoparent=1
```

Test, test, test and if you run into issue, you can move the URL back until you figure them out.

Signature Request Buttons

If you are not yet using Lightning, you can still create a Signature Request using a Visualforce page with a URL button. Please note that while this will technically work with Lightning and Mobile, it is not recommended.

1) Make sure you are in the "Classic" web version of Salesforce, with Admin permissions.

2) At the top-right of the page click on Setup → Customize → Opportunities → Buttons, Links and Actions, and click the **New Button or Link** button.

3) Give it a Name and Label like "Request Signature", set the Display Type as "Detail Page Button", set the Behavior to "Display in existing window without sidebar or header" and make sure the Content Source is URL.

4) In the main code area below, paste the following:

```
/apex/signature__RequestSignature?Id={!Opportunity.Id}
```

5) You will need to add to the URL using the parameters listed on Page 16. Each parameter should be separated by an ampersand (&) the same as the examples on that same page.

   ** Please note that the "id" parameter when using this URL is the same as the "parentid" parameter that is used for Signatures on that page.*

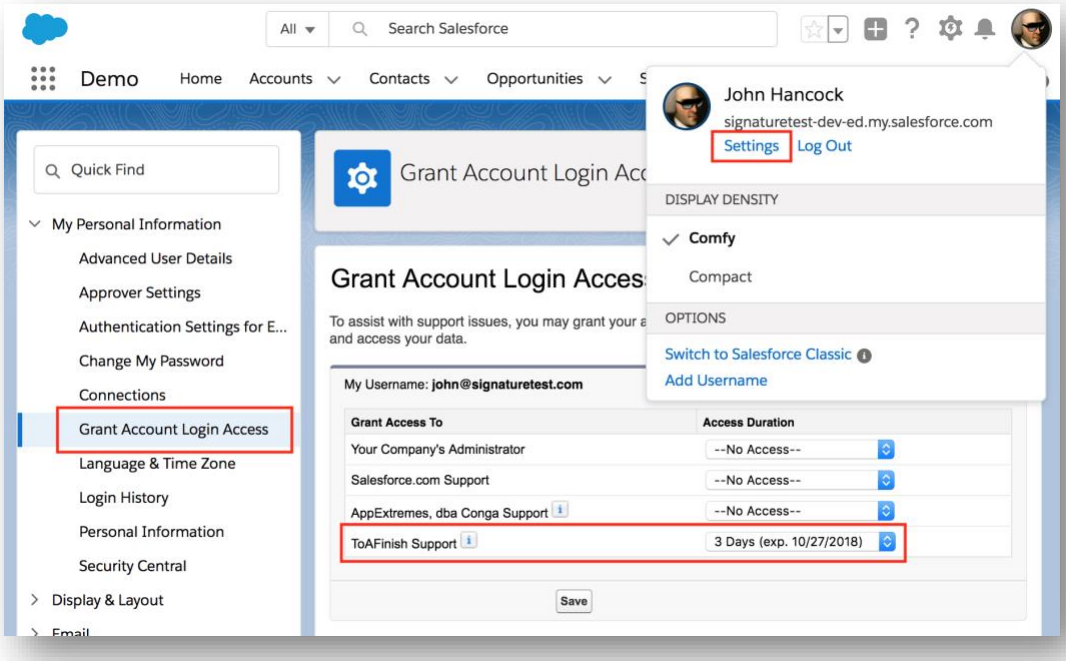
6) Save the button, and now go to Setup → Customize → Opportunities → Page Layouts and add it into one or more page layouts in order to be able to request signatures from there.

Now you must set up a Site where your contacts can sign the form you set up.

Congratulations on setting up the Signature Request process, now on to the hardest part, generating a custom PDF document for signature!

Feedback and Remote Access

Please send us your feedback to support@toafinish.com, and don't hesitate to email if you have questions about these instructions. When you do, you might want to give us remote access, as shown in the screenshot below, so that we can log in as you to see the problem.



Rating Simple Signature

We greatly enjoy being able to provide users with and app they will find useful. So, if you like the app, please leave us a positive review on the AppExchange:

https://appexchange.salesforce.com/listingDetail?listingId=a0N3A00000DqCoUUAV

But if you run into issues, *please* don't leave them there for the world to see (unless we can't get it resolved). Rather, contact us at support@toafinish.com and we'll work together to make this app To A Finish™.

Thank you!

The To A Finish™ team.