



SIMPLE SIGNATURE

IMPLEMENTATION GUIDE



TO A **FINISH**

SIMPLE SIGNATURE FOR SALESFORCE

Implementation Guide

© 2016-2026 To A Finish LLC
12407 N MoPac Expwy • Suite 250
Austin, TX 78758

www.SimpleSignatureApp.com

Table of Contents

1 – Introduction	5
Introducing To A Finish	6
Introducing Simple Signature	7
Free vs. Paid App Comparison	7
Continual Development – What It Means	8
Before You Begin, Requirements	8
2 – Basic Configuration	9
Permission Set Setup	10
Salesforce Settings Updates	10
Modifying the Signature Form	11
Adding Fields to the Signature Form	13
Connecting to a Parent Object	13
Launching a Signature Form	13
Launching from the Parent Object	14
Adding the Signature Button to a Layout	15
Adding the Signature Form to Salesforce Mobile	16
3 – Paid Options	17
Custom Signature Forms	17
Advanced Parameters	18
Creating a Signature Experience	22
Creating a Lightning Signature Form	23
Creating a Fullscreen Signature Form	24
Adding the Signature Form to a Lightning Page	25
Advanced Visual Changes	27
Aesthetic and Branding Changes	28
eSignature Agreement	29
Dynamic Form Changes	29
Expanded Custom Labels	31
Flow Plugin	32
Advanced Flow Options	33
Automated Parent Records Updates	38
4 – Signature Requests (paid)	40
PDF Document Creation	41

Site Setup	41
App Configuration	44
Signature Requests Custom Pages	49
Launching the Signature Request	50
Actions	50
Lightning Components	51
Email Template Setup	52
Testing	52
Advanced Features	53
Send From Emails	53
Email SPAM and Other Problems	54
Multiple Step Signatures	54
Signature Site and Data Security	59
Signature URL Security	61
User Countersignature	63
5 – Document Generation (paid)	64
Google Docs PDF Document Creation	64
Example Merge Codes	67
Related List	69
Dynamic Sections	71
Visualforce PDF Document Creation	72
Integration with other Document Generators	74
6 – Administration (paid)	76
Signature Maintenance	76
Signature Report/Bulk Printing	78
Multiple Salesforce Orgs	79
Sandbox Configuration	79
Production Deployments	81
Feedback and Support Access	83
Rating Simple Signature	83

1 – Introduction

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.

Thank you for considering our app in your organization.

I C O N K E Y	
	Details You Can Skip
	Don't Miss This!
	Sample Code
	Danger!

To succeed in the installation process, you will need to pay close attention to all of the information contained in this manual. We recommend reading through this first chapter before you begin, and then reading through each section as you work on it.

The Icon Key to the left shows the icons you will be seeing as you go through this manual. The more advanced readers can skip some of the details as indicated by the appropriate icon.

This manual is a work in progress, so please do take notes and let us know if you run into any issues with the text or if you believe there is a better way to accomplish the tasks we describe.

Introducing To A Finish



To A Finish™ is a concept company that has grown out of a desire to improve the Salesforce CRM consulting industry. Too often we have found the priorities of consulting companies skewed towards speed and profitability, rather than excellence and long-term success. This is not a blanket statement about all consulting companies, rather a general statement of the industry at large.

We believe there is a different way of consulting that will bring ultimate success to any CRM venture. The biggest difference is time. We give special attention to steps that are often rushed or even overlooked entirely in a typical project. Things like in-depth analysis, getting to really know a business, documentation and end-user involvement in the process help us set up a project for success. Ultimately, we must spend the time to do the work until it is absolutely complete; To A Finish™.

We are not guaranteeing our clients will never see a bug or have to rethink initial strategies. Reworking is actually part of the process of achieving success. What we are saying is that our strategies and focus will minimize unforeseen problems and achieve for us an ultimately successful project completion. We put our name on our work and stake our reputation on quality service.

Test us. Allow us to show you the difference of doing work To A Finish™.

You can find more about To A Finish at www.toafinish.com



TO A FINISH

When you finish a thing you ought to be able to say to yourself: "There, I am willing to stand for that piece of work. It is not pretty well done; it is done as well as I can do it; done to a complete finish. I will stand for that. I am willing to be judged by it."

- O. S. Marden

Introducing Simple Signature



Introducing a simple solution to capture signatures on the Salesforce platform. With Simple Signature you can acquire signatures on release forms, petitions, waivers, and other simple forms that need a few key data elements and a signature. Simple.

With a little implementation work, Simple Signature *can* be used to effectively replace other document generation apps for Salesforce at a much lesser cost. However, primarily it is a tool to quickly gather signatures, unencumbered by multiple-step processes or expensive, custom solutions. No need to create a Lead or Contact record in advance; just read, enter your name and maybe a couple other things, and sign. Keep it simple. Get it done.



Free vs. Paid App Comparison

The Paid version of Simple Signature includes many advantages. See a chart comparing the versions below:

Feature	Free	Paid
Unlimited, In-Person Signatures	✓	✓
Technical Support		✓
Field changes on the Signature Form	Limited	✓
Custom/Multiple Signature Forms		✓
Link to Parent Object	Limited	✓
Document Generation using Google Docs		✓
Signature Requests via Email & QR Code		✓
Final PDF Generation, saved to Files & Emailed		✓
Automated Lead (or other record) Creation		✓
Additional Signature Options (Typed, Paper, Topaz, None)		✓
Integration with 3 rd Party Apps		✓

Keep it Simple. Get it done.

SIMPLE SIGNATURE FOR SALESFORCE

Continual Development – What It Means

Simple Signature is a ... simple app, so not much can go wrong, but still, we know that somebody is going to find something that is broken. We are constantly adding features, and likely just made some pretty huge updates recently. So, please be gracious, and tell us about anything you run into. Email at support@toafinish.com and we'll gladly take a look at what you find, even if you are on the free version (bugs only).

We greatly enjoy being able to provide users with an app they will find useful. And a few of those users sometimes need us to build other things for them, so having a free version is one of the ways we showcase our work for potential clients.

So, if you like the app, please leave us a positive review on the AppExchange:

<https://appexchange.salesforce.com/listingDetail?listingId=a0N3A00000DqCoUUAU>

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™.

Before You Begin, Requirements



Before you begin working with our app, you must have several components already in place and configured. We will not cover these components in detail, but will at least point you to where you can get the information you need.

A production system Salesforce.com Org will be needed in order to set up the integration. A sandbox will be very helpful in this process as well. Any edition, Professional and above, of Salesforce.com should work.

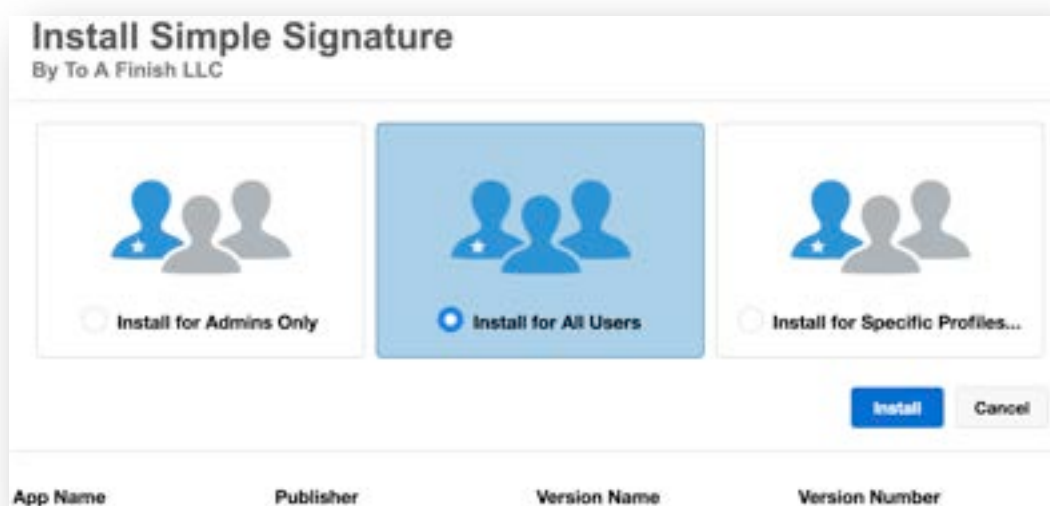
Salesforce.com

- ☞ Sales Cloud
- ☞ Service Cloud
- ☞ Custom Cloud (Force.com)

2 – Basic Configuration

The steps necessary to set up the app for full use

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.



When prompted, please approve third-party website access. This will allow you to use the various integrations that are built into Simple Signature.

If you do not do this step when you install, you might have to go to Setup later and look up “Trusted URLs” and “Remote Site Settings” to activate these website connections manually.

Permission Set Setup

In order to have access to Simple Signature, users will need to be granted the Simple Signature User permission set. In the Setup area go to Users → Permission Sets and then click on the **Simple Signature (Admin)** permission set.

Now, click the “Manage Assignments” button and add all Administrative users who will be accessing Simple Signature to this permission set. Do the same for regular users who need access with the **Simple Signature (User)** permission set.

ß

As you go through the configuration, you will often see this character. Some users confuse it with a B. However, it is a character from the German alphabet generally called the “sharp S” - optionally written as “ss” - our app initials.

Salesforce Settings Updates

Following is a list of recommended changes in the overall Salesforce settings. These settings may not be absolutely necessary, depending on which parts of Simple Signature are being used. We recommend that a Salesforce Administrator reviews this list before you make any changes, in case one of these settings will conflict with something else in the org.

Setting	Set to	Location	Description
Adopt updated CSP directives	False	Setup → Security → Session Settings	This removes the ability to edit and preview Google Doc templates for some browsers, due to the page configuration.
File Handling Options (.pdf)	Execute in Browser	Setup → Security → File Upload and Download Security	This is for the signing page on Signature Requests with embedded PDFs. If using a different setting, it would try to download the PDF instead of displaying it.
Use Lightning Web Security for Lightning web components and Aura components	True	Setup → Security → Session Settings	If this is not on, certain places with our signature forms will give an error when first loaded, though they might still function correctly.

Modifying the Signature Form

Now, before beginning to use Simple Signature, you will want to change what the form says in order to configure it to work like you need it to.

- 1) Make sure you are in the Lightning experience desktop version of Salesforce, with Admin permissions.
- 2) At the top-right of the page click on Setup → User Interface → Custom Labels.



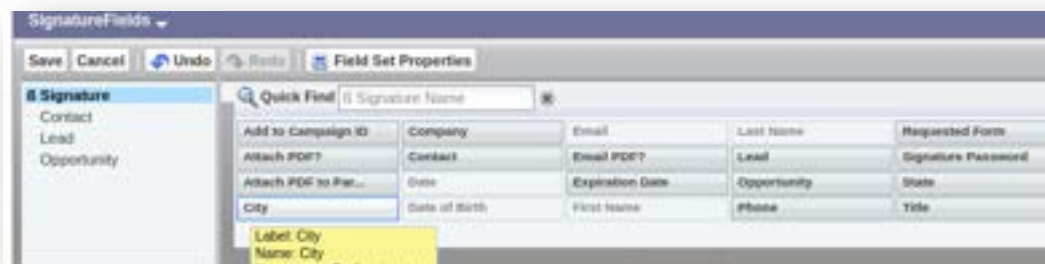
- 3) Find and then click on the Name of the **Page_Title** custom label.
- 4) You will then be presented with a page where you can create a “Translation” for this particular item. To do so, press the **New Local Translations / Overrides** button.



* Note: If you do not see the **Translations** area shown in the screenshot above, go to Setup → User Interface → Translation Workbench → Translation Language Settings, and turn it on. You should choose “English” or the native language of the users who will be using the form.

SIMPLE SIGNATURE FOR SALESFORCE

- 5) Select a Language, typically “English” or the default language your users will be using, and then type in the Title you want at the top of the form. Press the **Save** button, then click on the “Master Label” link, and you will see that this title is now a part of the form.
- 6) If you have users with Salesforce set to a different language, you will need to repeat steps 4-5 for each of those languages. If you do not, then they will default to seeing the title “Salesforce.com, Inc. Release Form”.
- 7) Now, go back to Step 2 and this time click on the **Long Legal Text 1** item. This is the text that will go under the title of the form, and before the fields and signature. If you can fit it all in, then go ahead and do so. You will also need to add text in the **Long Legal Text 2** item, even if it is only one character. Repeat Steps 4-6.
- 8) Now do the same Steps 3-6 for any other elements you wish to change on the form.
- 9) To remove fields from the form, go to Setup → Object Manager and then find and click on the **Signature** object.
- 10) Scroll down to the “Field Set” section and click “Edit” on the **SignatureFields** field set. You can now remove fields, reorder them, or add new fields to the form by dragging them down into the field set (see the next section for more).



- 11) Once you are done, go back to Salesforce Lightning and test the “Sign” tab, making sure that your entire wording is now as you wish it to be.

* Note: It is important to get this right before you start using the app, especially if you are using it for legal purposes. When somebody signs and saves the record, the wording they agreed to is copied onto the record, so it is important that the wording be right before you start taking signatures.

Adding Fields to the Signature Form

Using the steps above, you can add the fields that come with the form into the page. However, you can add as many custom fields to the form as you desire.

- 1) Go to Setup → Object Manager → Signature and click the **New** button to add fields. Repeat as necessary.
- 2) Once you've added all the fields you want, scroll down to the "Field Set" section and click "Edit" on the **SignatureFields** field set. You can now remove fields, reorder them, or add new fields to the form by dragging them down into the field set.

**Note: you can create a new Field Set if you prefer, but you will only be able to use it in the Paid version.*

- 3) Once you are done, go back to Salesforce Lightning and test the "Sign" tab, making sure that your entire wording is now as you wish it to be.

Connecting to a Parent Object

Signatures can be stand-alone, but you can also connect them to any object like Contracts, Orders, Cases or a custom object. If you do this, that object is considered the parent of the signature. You can either add a button on the parent object itself, or a button on the Signatures related list for the parent object, and the signature created will be a child of the parent object you selected.

Launching a Signature Form

In Salesforce we currently have several different technologies at play, all of which are supported in most cases in Simple Signature:

- 1) Lightning Web (which works with Lightning and Mobile), and
- 2) Lightning (which also works with Lightning and Mobile but a little slower),
- 3) Visualforce (which works with Lightning, Classic, Mobile and Mobile SDK). And is covered in a [separate document](#).

Lightning Web Components (LWC) is still being deployed by Salesforce, but development is moving in that direction, so we are committed to it as well. You should try to use LWC if you can, or if not Lightning if at all possible.

In the meantime, the only reasons to go with the older Visualforce offerings are in situations where your organization is still using Classic or if you have mobile users using the Mobile SDK. Virtually everything else is available in Lightning, and most is being released now in the new Lightning Web technology.

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 Mobile.

- 1) Make sure you are in the “Lightning” desktop version of Salesforce, with Admin permissions.
- 2) At the top of the page click on Setup → Object Manager and click on **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, if you are in the Free mode, skip this step. If paid, follow the instructions in the **Creating a Signature Experience** section on page 22. Then, follow the instructions in the **Creating a Lightning Signature Form** on page 24.
- 5) Once done with those tasks, go back to Setup → Object Manager and then click on the **Parent Object** you want to have the signature on, in this case **Contact**.
- 6) Click the **Buttons, Links and Actions** section, and then the **New Action** button. Choose to create an action to run a Lightning Component.
- 7) Select the Lightning Component that you just created in step #4 (or the **signature:SignatureFormFree** component if in Free mode), and enter a height at least 500px or 600px. Next give the Action a name such as “Sign”. The screen should look similar to below. Now **Save**.

SELECTING A PARENT FOR YOUR SIGNATURE

When you create a signature, Simple Signature will attempt to find your parent object based on where you are starting from, but sometimes you will want a different parent object, or for some reason we will not be able to connect to the parent object. If that happens, use some type of automation (such as a before flow or before trigger) to populate the Parent Lookup Field with the API name of the lookup field you WANT to use as the parent. Contact and Lead parent fields should automatically populate as well.



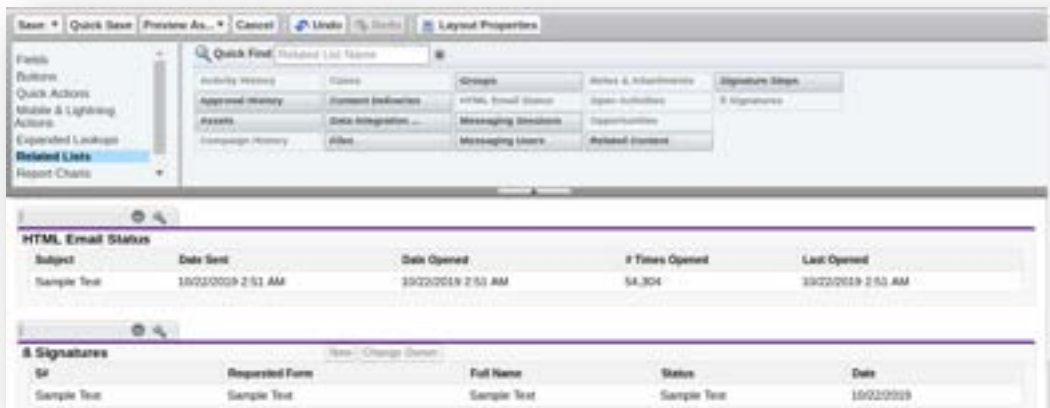
- 8) The next page explains adding the button to the layout.

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 “Mobile & Lightning Actions” and drag the action into the “Mobile & Lightning Actions” area.
- 3) We also recommend adding the “ Signatures” related list to your layout. But if you do, remove the “New” button so that users don’t get confused and try to add a signature through this button.



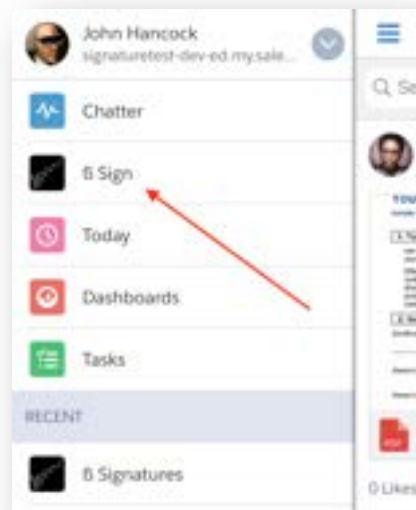
Adding the Signature Form to Salesforce Mobile

The first thing you may want to do is configure the Simple Signature form to show up on your Salesforce Mobile app.

- 1) Make sure you are in the Lightning experience on the web version of Salesforce, with Admin permissions.
- 2) Go to Setup → Apps → Mobile Apps → Salesforce → Salesforce Navigation.



- 3) Select “Sign” from the pages on the left, and use the arrow key to move it to the items on the right. Position it near the top so that it isn’t lost.
- 4) Press the **Save** button.
- 5) Now log into Salesforce and verify that you can, in fact, see the “Sign” tab. (You might need to close the app and open it again in order to see it.)
- 6) Test the process as instructed in the next chapter.
- 7) Next, go back to the Lightning desktop version of Salesforce to continue configuring.



You are done! Happy Signing!

3 – Paid Options

Optional setup steps for changing the signature experience

Simple gathering of signatures is available in the free version, but it must always include the Simple Signature branding when doing so. Once you upgrade to the paid offering, you are able to brand the signature form to meet your desired user experience. Things you can do include:

- 1) Changing the logo to your own logo.
- 2) Removing the “Simple Signature” branding (if desired).
- 3) Advanced coloring and HTML formatting on the page to match company guidelines.

These things are discussed in various areas when custom Visualforce pages, or Lightning components are offered. Doing them is fairly easy, once the paid version is in place. Keep reading for a few more areas that you can change.

Custom Signature Forms

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



* Note: A fairly advanced user will be able to follow these instructions to replicate the standard functionality using a custom action and Lightning components, but adding some 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.

Advanced Parameters

The **Paid Version** of Simple Signature supports many parameters that can be passed via the Signature Experience. A list of supported parameters is shown below:

Parameter	Description	Type
allowrequest	Will add a "Signature Request" button on the Signature Form under certain conditions, which when used, creates the Signature record with a "Requested" status, unchecked/False (default) to do nothing. Options are: • In-Person Signature • Signature Request Only (default) • Signature Request and In-Person Signature	Requests
attachpdf	Checked to attach the PDF generated to the Signature record, Unchecked (default) to do nothing.	Universal
attachpdftoparent	Checked to attach the PDF generated to the Parent record, Unchecked (default) to do nothing.	Universal
autoagree	When the "submitttype" parameter is used and is set to "None", then the signature pad is not shown, and instead there is a checkbox. If this parameter is checked, it should automatically allow the user to save without seeing this checkbox.	Advanced
backtoparent	Checked to go back to the parent record when done signing , Unchecked (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.	Advanced
compact	Checked to shrink the fields on the signature form so that each label is on the same line as the field, Unchecked (default) the labels will be above the field, allowing more room for data entry.	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
country	Used with advanced code. Please contact technical support to use.	Advanced
defaultfilterfield_	Select a Lookup field from the Signature object in the "Signature Field" column, and then enter API Name of the field in the Lookup object that said lookup field will be filtered on. (Use with the defaultfiltervalue_ parameter.)	Advanced
defaultfilterlookupvalue_	Please contact technical support to use this advanced feature. Currently works only with Visualforce.	Advanced
defaultfiltervalue_	Select a Lookup field from the Signature object in the "Signature Field" column, and then enter value which the lookup field will be filtered on using the Lookup object field indicated in the defaultfilterfield_ parameter.	Advanced
defaultvalue_	The "defaultvalue_" parameter is dynamic, next you must pick the name of a field in the Signature object. This selected field will be populated with the value you provide for the parameter.	Universal
deletesigantureafter	Checked to delete the PNG image of the signature after the final PDF is generated, Unchecked (default) to do nothing.	Advanced
emailpdf	Checked to email the PDF generated to the contact attached, Unchecked (default) to do nothing.	Universal

SIMPLE SIGNATURE FOR SALESFORCE

Parameter	Description	Type
emailtemplatecompletion	The name (not the label) of the email template to be used for the email that goes out after the signature request is processed successfully.	Advanced
emailtemplaterequest	The name (not the label) of the email template to be used when sending Signature requests.	Advanced
emailWithExternalLimit	By default, signature requests will attach themselves to an existing lead or contact with the email address provided. If one is not found, a temporary contact is created in order to send the email using standard Salesforce email processes. However, this option, when checked, will instead send via a limited number of unattached emails (currently Salesforce limits to 5000 per day) and will not attempt to attach itself to a lead or contact. Note: using this option will prevent Activities from being created for the signature requests.	Advanced
embedpdf	Checked to show the PDF to be signed in an iFrame above the signature form at the time of signing, Unchecked (default) to show the form by itself.	Universal
fieldheader	API name of the field(s) to show as Headers on the signature form. Separate multiple fields with commas.	Universal
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	Field Set to use on the Signature Form. Uses SignatureFields as the default if not included.	Universal
formtype	Enter the name of the Form Type to be used. Form Types are useful for managing multiple signature request types, each with its own Signature and Thank You pages as well as Legal Text. For instructions on using form types, please check the XXX section.	Advanced
generateqrcode	Checked to generate a QR Code image and save it in the QR Code Image field, and ALSO to show the QR Code image after a signature request, Unchecked (default) to do nothing.	Requests
hideblankfields	Similar to “hideblanks” but only hides blank fields entered in the “Hardcoded Value” column. Enter one or more API field names, separated by a comma.	Advanced
hideblanks	Checked to hide the input fields if they are blank, Unchecked (default) to show all fields on the signature form.	Universal
hidecancel	Checked to hide the Cancel button on the Signature Form, Unchecked (default) to do nothing.	Universal
language	Language of the form. Only works with the Visualforce components.	Universal
lookupdisplayfields_	Select a Lookup field from the Signature object in the “Signature Field” column, and then enter List of fields (API names), separated by commas to include in the lookup when used on the form.	Advanced
lookupreturnfield_	Select a Lookup field from the Signature object in the “Signature Field” column, and then enter List of fields (API names), separated by commas to pull in the query when processing the form. These fields will not be visible (contrary to the “lookupdisplayfields” parameter above), but will be available in code, for advanced processing.	Advanced
multisignature	Enter the names of up to 5 separate signature pads on the same signature form, separated by commas. Remove parameter to leave the default of only 1 signature pad.	Universal

SIMPLE SIGNATURE FOR SALESFORCE

Parameter	Description	Type
	The first signature pad on the list is automatically required, the others optional by default. Put an asterisk at the beginning of the name to make it required. In this example, the first, fourth and fifth signature pads would be required: "One,Two,Three,*Four,*Five"	
multistep	Checked to create a Step record when the Signature request is created, Unchecked (default) to remain a single Signature request.	Requests
nosignature	Checked to allow submitting the form without signing, Unchecked (default) to require signature.	Requests
optionaluploads	Enter the names of up to 5 separate images that may be uploaded when signing. For example, entering the following (without the quotes): "Drivers_License,Passport,Library_Card" Would put 3 separate upload elements between the form fields, and the signature area below. The signer would be able to select an image for each of these items and they would be uploaded and saved as files in the signature record, upon submission. If Rich Text fields with API names matching the names of the uploads required are available in the Signature object, these fields will also be populated with a link to the image, allowing them to be shown on the record detail or merged into documents. In the case above, the fields would need to be named: Drivers_License__c, Passport__c and Library_Card__c.	Advanced
parentid	The Id of the record to be considered the parent of the Signature. Note that this is not needed if you are launching the Signature from the Parent record.	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
preauthorized	Checked to hide the eSignature Agreement prompt, if this is turned on. Unchecked, it will show the eSignature Agreement popup if it is configured to do so.	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
requestedpdfnamepreview	Checkbox option, which when turned on, will also name the preliminary PDF generated to send as an attachment in the signature request email, with the name assigned using the requestedpdfname parameter (see above), but with "_Preview" appended at the end of the filename. (Google Docs only)	Requests
requireduploads	Works the same as the "optionaluploads" parameter, but these are required before the signer is allowed to submit the form. If included, these upload components will be shown above the optional uploads on the form.	Advanced
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. Can be used to enforce a custom page at the end of the signature request process. For example, a value of "CustomThanks" would send the signee to a	Requests

SIMPLE SIGNATURE FOR SALESFORCE

Parameter	Description	Type												
	VisualForce page on the Site with this name after they sign. <i>* Code for a custom thank-you page can be found in our GitHub repository.</i>													
showattachmentid	The Salesforce Record ID of an Attachment to show on a new Signature Form. This allows displaying a PDF on a Signature that is not yet created.	Advanced												
submissiontype	Special parameter, allowing a user to select one or more ways that signees can sign the form. Allowed types are: <table><tr><th>Option</th><th>How the User would Sign...</th></tr><tr><td>Regular</td><td>(Default) User can sign with a mouse or their finger</td></tr><tr><td>Typed</td><td>User can type their name and select a font to use as a signature</td></tr><tr><td>None</td><td>User only checks a box and does not need to sign</td></tr><tr><td>Paper</td><td>User prints out the PDF or Signature Form and physically signs (a “wet” signature), then scans, and uploads the document – please note that with this option, the final Status of the Signature record will be “Uploaded” and needs a Flow to change it to “Signed” after whatever vetting is deemed necessary by the business (there is no way to verify that the document uploaded is actually the required document signed.)</td></tr><tr><td>Topaz</td><td>User signs on a Topaz Signature Pad only (requires device). Please note that the Topaz SDK is actively being developed by the makers, and new features are being introduced all the time. This adds a bit of inconsistency at times, so we consider this function to still be in BETA mode. Also, please note that you will need to go to Setup → My Domain and Edit the main settings to ensure that the “Stabilize Visualforce, Experience Builder, Site.com Studio, and content file URLs” new option is turned on. If not, the Topaz popup will be blocked as Salesforce would see it as a different site.</td></tr></table>	Option	How the User would Sign...	Regular	(Default) User can sign with a mouse or their finger	Typed	User can type their name and select a font to use as a signature	None	User only checks a box and does not need to sign	Paper	User prints out the PDF or Signature Form and physically signs (a “wet” signature), then scans, and uploads the document – please note that with this option, the final Status of the Signature record will be “Uploaded” and needs a Flow to change it to “Signed” after whatever vetting is deemed necessary by the business (there is no way to verify that the document uploaded is actually the required document signed.)	Topaz	User signs on a Topaz Signature Pad only (requires device). Please note that the Topaz SDK is actively being developed by the makers, and new features are being introduced all the time. This adds a bit of inconsistency at times, so we consider this function to still be in BETA mode. Also, please note that you will need to go to Setup → My Domain and Edit the main settings to ensure that the “Stabilize Visualforce, Experience Builder, Site.com Studio, and content file URLs” new option is turned on. If not, the Topaz popup will be blocked as Salesforce would see it as a different site.	Advanced
Option	How the User would Sign...													
Regular	(Default) User can sign with a mouse or their finger													
Typed	User can type their name and select a font to use as a signature													
None	User only checks a box and does not need to sign													
Paper	User prints out the PDF or Signature Form and physically signs (a “wet” signature), then scans, and uploads the document – please note that with this option, the final Status of the Signature record will be “Uploaded” and needs a Flow to change it to “Signed” after whatever vetting is deemed necessary by the business (there is no way to verify that the document uploaded is actually the required document signed.)													
Topaz	User signs on a Topaz Signature Pad only (requires device). Please note that the Topaz SDK is actively being developed by the makers, and new features are being introduced all the time. This adds a bit of inconsistency at times, so we consider this function to still be in BETA mode. Also, please note that you will need to go to Setup → My Domain and Edit the main settings to ensure that the “Stabilize Visualforce, Experience Builder, Site.com Studio, and content file URLs” new option is turned on. If not, the Topaz popup will be blocked as Salesforce would see it as a different site.													
submissiontypedefault	If multiple types are selected with the submissiontype parameter above, this parameter allows a default tab to be set.	Advanced												
submissiontypefonts	Works with the “Typed” type when using the submissiontype parameter, above. Specific fonts can be included, separated by commas. For example, if “Kristi,Meddon,Pacifico” is entered as the value for the parameter, only these three fonts would show up when signing. <i>* Only existing fonts can be used, new fonts cannot be added by typing a desired font name, sadly.</i>	Advanced												
submitbutton	Will change the label of the Submit button on the Signature Form. This button label can also be changed with custom labels, but this parameter allows you to have different submit button labels for different signature experiences.	Advanced												
templates	ID(s) of the Template record to generate for signature, and at the end of the signature process. Separate multiple IDs with a comma.	Universal												

Deprecated parameters:

- sdocstype
- optionimagename
- parameterfield
- optionpopulatedfrom
- signaturebox
- optionsavedin
- twosignatures
- donotpreload

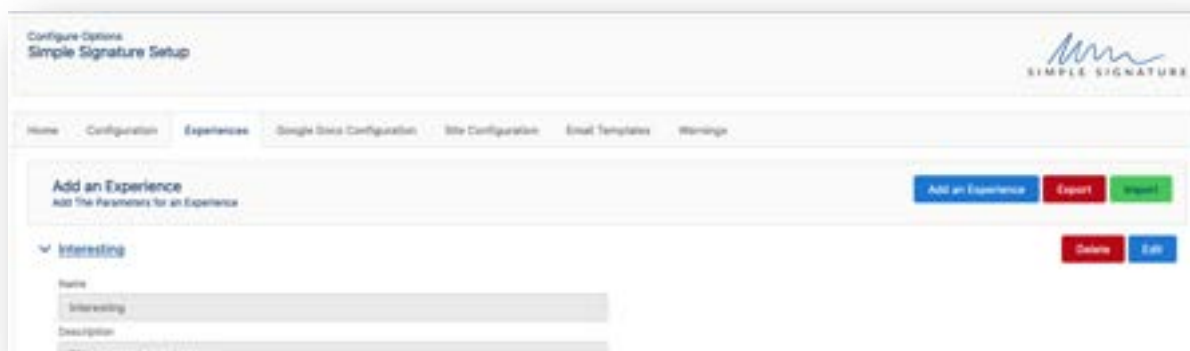
Creating a Signature Experience

Lightning components need parameters to determine how the signature form will behave. This is done via setting up an Experience. This experience can contain as many parameters as needed and will generate a unique Experience Id, which you can plug into a Lightning component in order to specify how that Lightning component should work:

- 1) In Lightning, go to the Simple Signature app and click on the **Configure** tab.



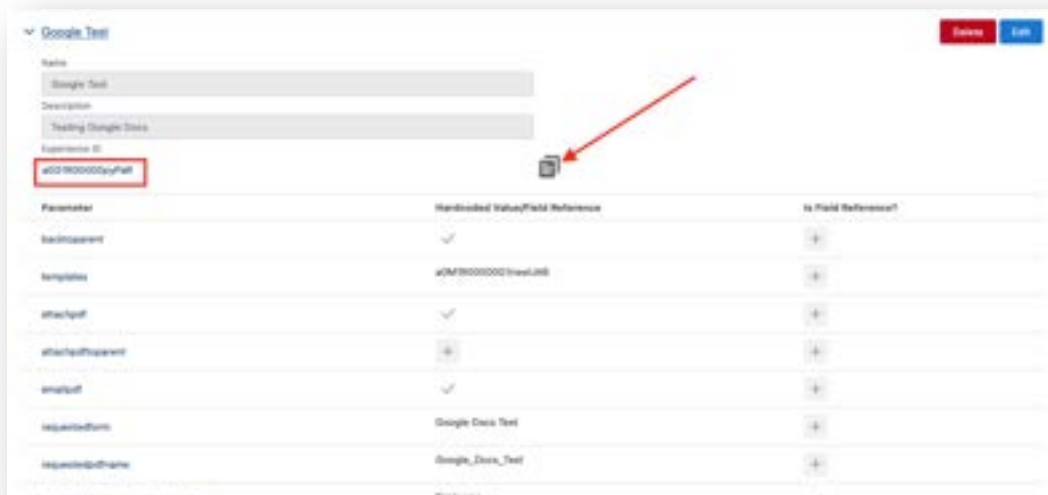
- 2) Click on the **Experiences** tab and then click the **Add an Experience** button:



- 3) Fill in the Name of the experience, and optionally a Description, and then click on the **Add Parameter** button to start with the configuration.

** The **Add Advanced Parameter** button does the same thing, but only brings up the advanced parameters, which are rarely used.*

- 4) Continue to add parameters listed on page 18 until you are done. Remember to press the **Save** button at the bottom when you have added all necessary parameters:



5) Now copy the “Experience ID” value to be used in a Lightning Component by pressing the **Copy** icon.

Creating a Lightning Signature Form

In order to use Simple Signature in Lightning and Mobile, a Lightning component must be used. This is only available in the Paid version. A generic component can be created to be used in multiple objects, or a more specific one can be built using an Experience ID (see the next section) to specify how it should act.

- 1) To create a new Lightning Component for the signature form, click on  at the top-right of the page and choose **Developer Console**.
- 2) In the Developer Console, click File → New → Lightning Component and then enter a name similar to “Sign_Custom_Object” and create the component.
- 3) This should take you to a blank, new component, in the “Component” tab. You want to add the code below, called “StandaloneSignatureForm-Simple.html” in the “Lightning” folder in our GitHub repository.

LWC > Aura

For the more technically minded readers, please use the LWC option in our GitHub repository, rather than the Aura component we recommend here. LWC runs faster and Salesforce recommends we use these components moving forward. The name of the Lighting Web Component is:

StandaloneSignatureFormLWC-Simple.html

GitHub.com Simple Signature Sample Code Repository

<https://github.com/ToAFinish/simplesignature>



```
<aura:component implements="force:appHostable,
    force:hasRecordId, force:lightningQuickAction" >

    ... MORE CONTENT HERE ...

    <signature:SignatureJunction recordId="{!v.recordId}"
        ExperienceID="XXXXXXXXXXXXXXXXX"/>

    ... MORE CONTENT HERE ...

</aura:component>
```

- 4) In the sample code above (the real code is in GitHub) you will see a spot to put the **Experience Id** from the prior section. Copy the value and paste it into the code, making sure there are not blank spaces between the quote marks.
- 5) Now press **Ctrl+S** to save the component, and close it out if desired.

*The **Launching from the Parent Object** section and the **Adding the Signature Button to a Layout** section starting on page 15, explain how to create an Action that will launch this component.*

Creating a Fullscreen Signature Form

The Lightning Component above will work as a popup in Lightning, or as a full-screen page in Mobile. If you want to have full-screen experience in Lightning, please use the sample code in the same folder to build the component: **Fullscreen_Component.html** and **Fullscreen_controller.js** will get you started.

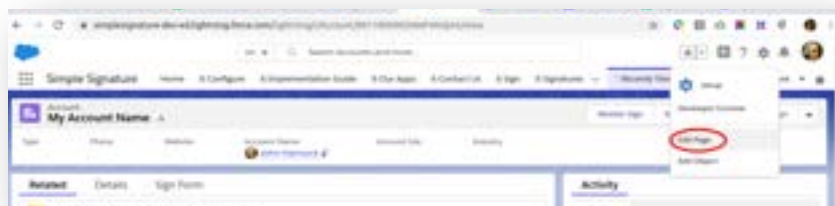
Do not hesitate to contact technical support at support@toafinish.com for help with setting up a full screen signature form if you run into any issues.

Adding the Signature Form to a Lightning Page

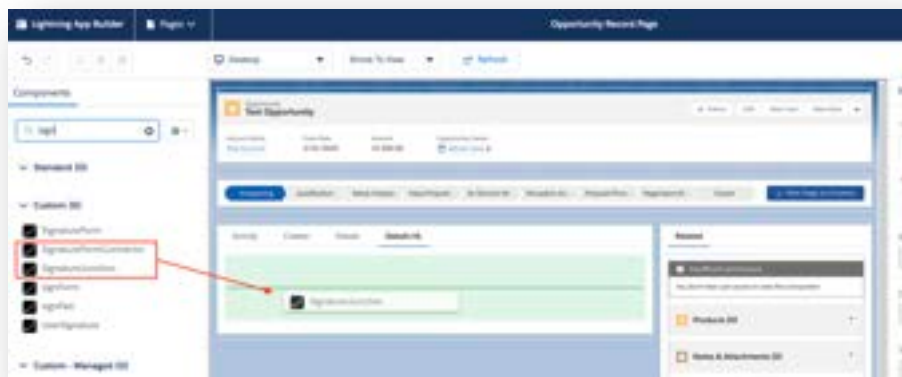
In addition to running the Lightning component through an Action, the form can be embedded into a Lightning page so that it is available for anyone with access to the record and access to Simple Signature to run. Here are the instructions to do so:

**Note: Before you can add a Signature Form to a Lightning Page, a lookup from the Signature object must be created, pointing to the current object. See page 14 for instructions on how to set this up.*

- 1) In the “Salesforce Lightning” web version of Salesforce, with Admin permissions.
- 2) Navigate to any record page of the object where you want a signature. Click on the  icon at the top right corner. Click on **Edit Page** to go into the Lightning page editor.

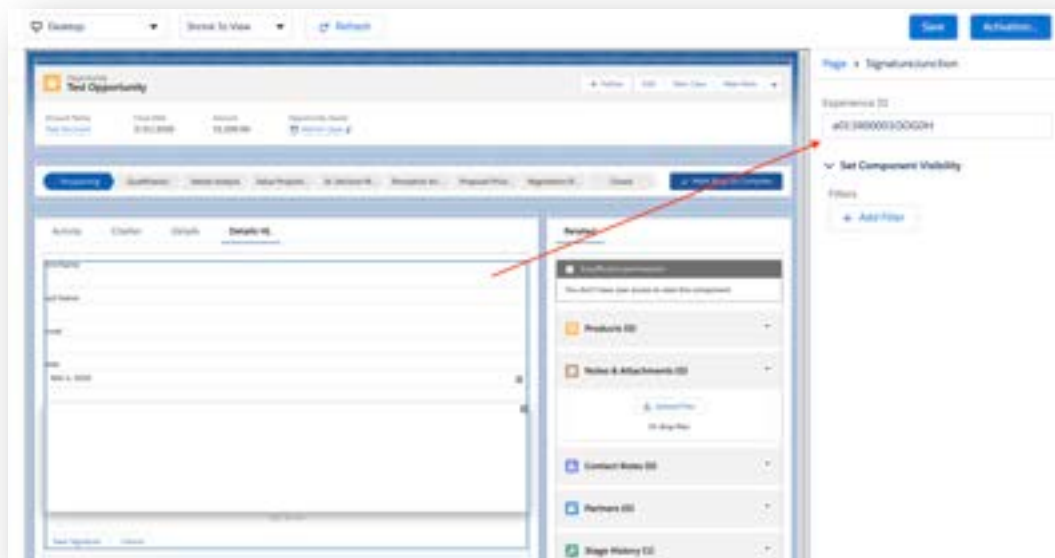


- 3) You will now be in Lightning App Builder. Determine where you want to put the signature form. Typically, you would want to add a new tab and give it a custom name such as “Sign” or “Signature Form.”
- 4) Drag the **SignatureFormConnector** component from the left panel into the new tab.



- 5) Now, click on the Lightning Component you just dragged in. You will see a blank parameter titled Experience ID. Please paste in the value from the Experience ID that you created starting on page 22.

SIMPLE SIGNATURE FOR SALESFORCE



- 6) Now **Save** the page and, if necessary, release it to the users.
- 7) Click the **← Back** button to return to the record where you started. Now you should see the Signature form on the new tab you created.

Advanced Visual Changes

Signature Forms can be changed in several different ways to appear different visually. See the chart below for different ways they can be changed and where to affect that change. Some of these changes can be done by updating certain fields on the Signature record, and if so, they are noted. Note that most of this can also be done by hard-coding the changes in a custom page:

Component	How Can This Be Changed?	API Field
Page Title	The “Page_Title” custom label can be changed. (page 11) The “formtype” parameter can be used. (page 30)	N/A
Page Logo	The “Logo Static Resource” and “Logo Style” elements can be updated in the Configure page. (page 29)	N/A
Legal Text	The “Long_Legal_Text_1 / 2” custom labels can be changed. (page 11) The “formtype” parameter can be used. (page 30)	signature__Legal_Text__c
PDF	The “templates” parameter can be used with Google Docs or VF pages. (page 22) The “pdftemplate” parameter can be used for VF pages only. (page 20) The “embedpdf” parameter must be used and set to True. (page 19)	signature__Templates__c signature__PDFTemplate__c signature__EmbedPDF__c
Fields	The “fieldset” and “compact” parameters can be used. (page 19) The “Field / Button Area” in the Configure page can be used. (page 29)	signature__FieldSet__c
Signature Type	The “submissiontype” and “submissiontypedefault” parameters can be set to allow one or more of these types of signatures for the form: standard, typed, paper, Topaz, None. (page 21) The “Signature Pad” and “Signature Top Border Left” and “Signature Top Border Right” elements can be updated in the Configure page. (page 29)	Signature__Submission_Type__c
Sign Button	The “Submit” and “Submit_and_New” and “Submit_Button” custom labels can be used to rename the Sign button. (page 11) The “submitbutton” parameter can be used to rename the Sign button. (page 22) The “Save Button” element on the Configure page can be used. (page 29)	N/A
Cancel Button	The “Cancel” custom label can be used to change the Cancel button. (page 11) The “Cancel Button” element on the Configure page can be used. (page 29)	N/A
Request Button	The “Request” custom label can be used to change the Request Signature button. (page 11) The “Request Button” element on the Configure page can be used. (page 29) The “allowrequest” parameter can be used to enable signature requests. (page 18)	N/A

** Many other parameters can be used to make the form act differently, in particular ones that set different rules before loading the actual form, or parameters that change the way fields are displayed.*

Aesthetic and Branding Changes

The Lightning signature forms have a blue and green color scheme that is customizable. This can be changed by going to Setup → Installed Packages → Simple Signature → Configure and on the “Configuration” tab, you will see the “Form Options”.

You can see the default CSS for each element and make any changes you want. However, please keep in mind that this is standard CSS and if you are not careful, it could break something. So, make small changes, test, and then repeat until you get it just right.

Use these options to change the colors of the form using CSS.	
Save Button (default = background:green;color:white;)	background:red;color:blue;
Request Button (default = background:blue;color:white;)	background:red;color:#000000;
Cancel Button (default = background:blue;color:white;)	background:brown;color:yellow;
Signature Top Border Left (default = height: 10px;background: blue;)	height: 0px;background: white;
Signature Top Border Right (default = height: 10px;background: green;)	height: 30px;background: pink;
Logo Static Resource (default = SSLogo)	MyLogo
Logo Style (default = position:absolute;top:0px:right:15px;)	position:absolute;top:10px:right:25px;
Field / Button Area	width:90%;margin: auto;
Signature Pad	width:75%;margin: auto;

** Note: the “Field / Button Area” and “Signature Pad” entries would allow you to potentially have the signature box float to the right of the fields and submission buttons, and it helpful especially when using Simple Signature in a language where a signature would be expected to be square-shaped.*

Don’t forget to press the green **Save Settings** button towards the bottom of the page!

eSignature Agreement

Some companies desire the user to agree to using eSignatures before actually signing. If you want to set that up, go to Setup → Installed Packages → Simple Signature → Configure and on the “Configuration” tab, you will see the “Form Options.” Turn on the “Require eSignature Agreement” option.



Dynamic Form Changes

A new parameter “formtype” is available which provides support for multiple different signature types in a single org, in particular for signature requests. By default, different signature forms can fairly easily be accomplished without this parameter as long as the signature types use the same legal text and signature site, but once you introduce variables such as differing legal text and a signature page that is different enough to necessitate different URLs, then the “formtype” parameter is what is needed. Set up steps:

- 1) First, go to Setup → Installed Packages → Simple Signature → Configure to the app configuration page, and click on the “Experiences” sub-tab. This will show you the list of signature experiences currently configured in the org. Expand the one you want to modify (or create a new one).
- 2) Click on the **Edit** button to modify your selected Signature Experience and press the **Add Advanced Parameter** button. A new parameter will be added to the list. Click on this parameter and select “formtype” and then enter a name for the form type in the “Hardcoded Value/Field Reference” column, and Press the **Save** button.
- 3) Now, go to Setup → Custom Code → Custom Metadata Types and click **Manage Records** next to the “Form Type” entry.
- 4) Create a new record with the same Name that you gave the parameter name in step #2. The other values in this form are as follows:

- a. Change the “Label” to be different from the “Name” if you want the page title itself to be changed on the signature page. If you leave it the same, then the Page_Title label value will be used.

- b. Change the “Signature URL” value if you want to have a different Site to use on this form, possibly due to a different VisualForce page you want to use for the Signature Form. Be sure to include the entire URL, including the “/” at the end. If this is blank, then the default page in the configuration will be used.

- c. Change the “Thank You URL” to set a URL where the signee should be taken to after signing. This is similar to using the “rurl” parameter, but includes the whole URL instead of just the page name.
- d. Change the “Legal Text” to use legal text that is different (perhaps longer) than what you have in the **Long_Legal_Text_1** and **Long_Legal_Text_2** custom labels.

- e. **Save** the record.

- 5) Now, go test the form to ensure that you are seeing the new legal text. If you are not seeing it, here are a few things to check:

- a. Ensure you are on the latest version of Simple Signature.
- b. Ensure the form is using the “SignatureFormInfo” component, and not referencing the legal text on the form directly. If this is not the case, you can go to our GitHub repository to copy code from one of the Standalone components in the “Lightning” folder.

GitHub.com Simple Signature Sample Code Repository

<https://github.com/ToAFinish/simplesignature>



- c. If the legal text is not showing on the signature page itself, please ensure you are using the latest code from the Custom_SiteSignature page in the “Pages” folder.

Expanded Custom Labels

Each signature form uses a field set to display fields. Each field within the field set has a label that is placed either to the left (default) or on top of (compact mode) the field. The length limit for these values is 40 characters. This is the length that is set by Salesforce and cannot be changed. But if you need to have longer labels on the signature form, Simple Signature provides an advanced method to add labels up to 255 characters in length. Here are the steps to set that up:

Go to Setup → Custom Code → Custom Settings → **Manage** on the “**Custom Labels**” custom setting. This will show you a list of actual expanded labels. You can create a new one by clicking the

Translations in a Site

In addition to field names, you can also use this space to translate actual Custom Labels.

Particularly useful in a Salesforce Site, where you are not allowed to set a language for a component embedded in the page, this custom setting allows you to change the labels on buttons and other places. Use the label API name instead of the field API name to achieve this. And remember, the language parameter must be explicitly set on the page URL.

Name	API Name	Label	Language Code
Testing01	Have_you_EVER_been_suspended__c	Have you EVER been suspended from or sanctioned by the Medicare, Medicaid, or any other program funded with federal, state, or local monies, or has your participation status in any such program EVER been modified (terminated, suspended, restricted, revoked)	EN

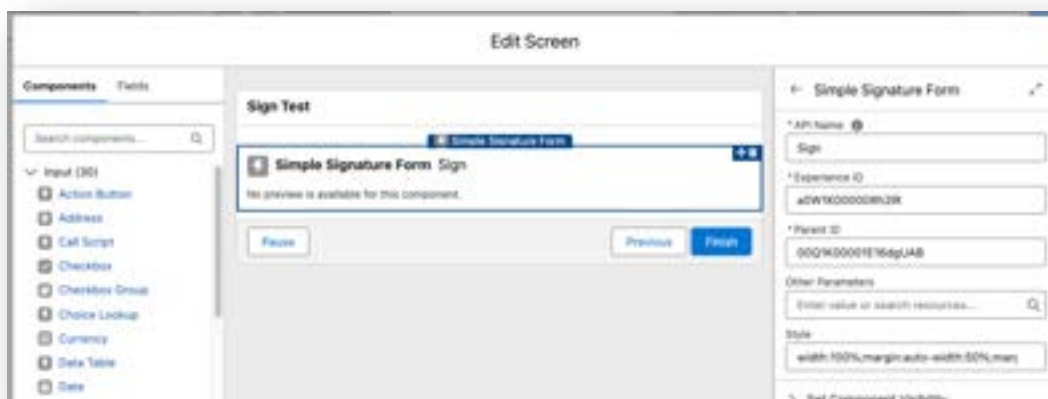
New button, and then populate each field:

- Give the label a unique “Name”, this is not used anywhere except here in this list.
- Populate the “API Name” with the actual API name for the field that you want to have an expanded label for. For example, if you want to update the Company field on the signature object, the value would be: “signature__Company__c”. *Please keep in mind that this will expand the label on all Simple Signature forms, so you cannot do this for only one form.*
- Populate the “Language Code” with the two-character language code being used on the form. This would typically be “EN” for English. *Please remember that you also need to use the “language” parameter in the Signature Experience with a matching value (e.g. “en” for English).*
- Populate the “Label” with a replacement expanded label with a maximum of 255 characters.
- And then **Save**. You can test any signature form with this field now, and it should use the expanded label.

Flow Plugin

A plugin can be used for collecting signatures in a Screen element from within a Flow. The signature screen can be put anywhere along the line in the process. Use these steps to set it up:

- 1) Go to Setup → Process Automation → Flows and click on the Screen Flow where you want to add the Signature process. You can edit an existing flow, or to test in a new flow, press the **New Flow** button. Select “Start from Scratch” and, select “Screen Flow” and press **Create**.
 - a. If you are creating a new flow, you will need a **record Id** to use as the parent record of the signature (we recommend a Lead record for now) as well as an **Experience Id**, which you will have set up following instructions in the *Creating a Signature Experience* in the prior chapter.
- 2) For a new flow, you can put the Signature in at the beginning, or if you are editing an existing one, determine where you want to add the Signature. Now, press the + Icon and select “Screen”.
- 3) This will now open a “New Screen” blank screen. Fill in a Label and API Name, and optionally, expand the “Configure Footer” section and uncheck the box to “show Footer” - this will restrict the use from leaving the signature area without signing.
- 4) Then, from the Components menu on the left, scroll all the way down until you see the “Simple Signature form” option. Drag this into the main area of the screen and click on it. Options will show up on the right.



- 5) Enter the parameters as explained below:
 - a. **API Name** – enter whatever makes sense
 - b. **Experience ID** – this needs to be the ID that is generated when you set up a Signature Experience, as explained in the prior chapter.

- c. **Parent ID** – this needs to be the record ID of a Salesforce record which will serve as the parent for the signature that will be created. So, if you start on a Lead, for example, the Parent ID should be the Lead ID field. If you start on a custom object, it would be the Id field of that custom object that is the “parent” to the Signature record.
- d. **Other Parameters** – this is a more advanced optional function, and can be used to pass the same parameters as used when setting up the Signature Experience. The most common use-case is to create the Signature prior to this screen, and then using the ID parameter to ensure this screen will be signing an existing Signature record, and not creating a new one. To do that you would type in something like: id=a0LKk000000CbL6MAK
- e. **Style** – this is optional and can be used with common CSS to ensure that the form displayed fits into the desired space. A common use is shown below, which would set the width of the screen to 100% of available space, and the width of the signature area to 50% of the available space: width:

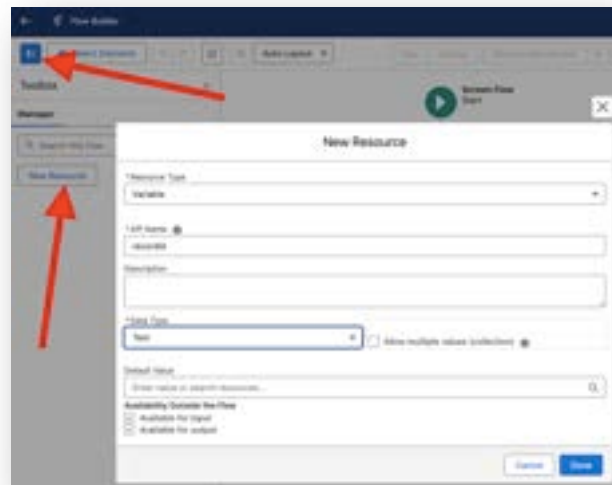
```
100%;margin:auto-width:50%;margin:auto
```

- 6) Now press the **Done** button, and then the **Save** button to save the Flow. You can then press the **Run** button to test it.

Advanced Flow Options

The previous section shows you how to add a signature to an existing flow. Now, we will create an advanced flow that can be used in real-world situations. It will use several features that need to be configured first, including Google Docs, which is covered in chapter 5. So, make sure you have configured Google Docs, created a template, and added it to the Signature Experience you will use in this Flow.

- 1) Go to Setup → Process Automation → Flows and click on the Screen Flow where you want to add the Signature process. You are creating a new flow, so, press the **New Flow** button. Select “Start from Scratch” and, select “Screen Flow” and press **Create**.
- 2) First, we need to create a variable to link to the parent Lead record. From the menu at the top, click the small Toolbox icon at the top-left, then press the **New Resource** button.
 - a. Select ‘Variable’ for the **Resource Type**.



SIMPLE SIGNATURE FOR SALESFORCE

- b. Give it this exact name for the **API Name**:
→ recordId ←
 - c. For the **Data Type**, select “Text”.
 - d. Select both “Available for input”, and “Available for Output” options, and press the **Done** button. If necessary, close the toolbox on the left.
- 3) Now, we need to incorporate the Parent record we will be using, in this case a Lead record. Press the + Icon and select “Get Records”. Fill it out so as to acquire the Lead record with all its fields:
 - a. Give it whatever **Label** and **API Name** you want
 - b. For the **Object**, select “Lead”
 - c. Give it only one **Filter**, where the “Id” field Equals the new variable you created in the last step.
 - d. All other options should be fine as they are by default.

The screenshot displays the Salesforce Flow Builder interface. On the left, a canvas shows a flow starting with a 'Start Screen Flow' icon, followed by a plus sign, then a 'Get Parent Lead Get Records' step (highlighted with a red box), another plus sign, and finally an 'End' icon. The right-hand pane shows the configuration for the 'Get Records' step.

Get Records

*Label: Get Parent Lead *API Name: Get_Parent_Lead

Description:

Get Records of This Object

*Object: Lead

Filter Lead Records

Condition Requirements: All Conditions Are Met (AND)

Field	Operator	Value
Id	Equals	{!recordId}

+ Add Condition

Sort Lead Records

Sort Order: Not Sorted ⚠ If you store only the first record, filter by a unique field, such as ID.

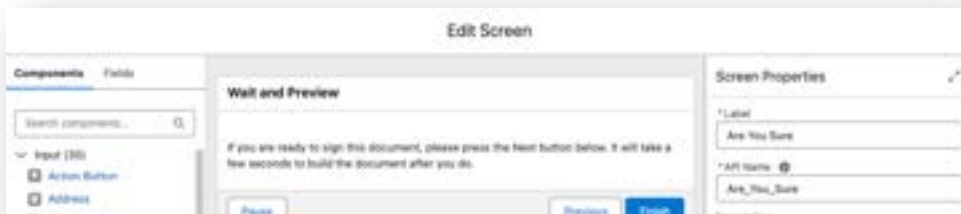
How Many Records to Store

☒ Only the first record
☐ All records

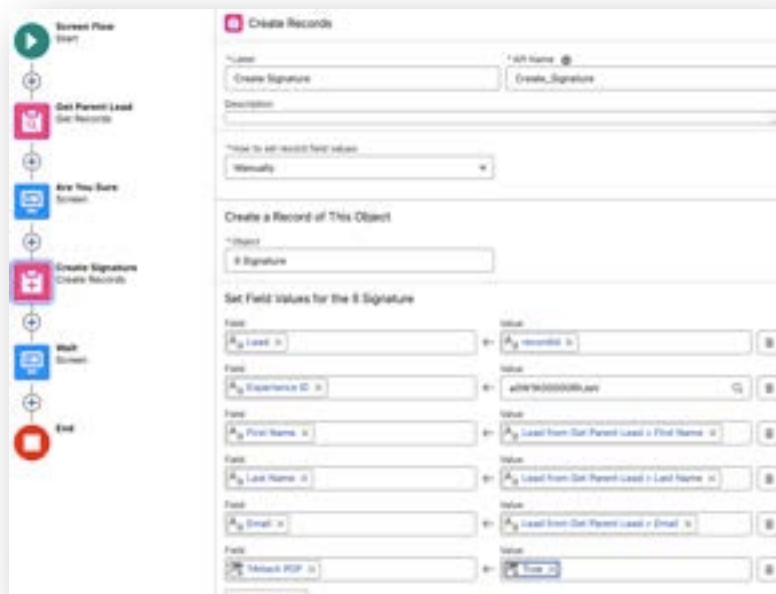
How to Store Record Data

☒ Automatically store all fields
☐ Choose fields and let Salesforce do the rest

- 4) Now, we want to place a confirmation screen so the user knows what is going on. Press the + Icon again, select “Screen” and populate it something along the lines of the screenshot below.

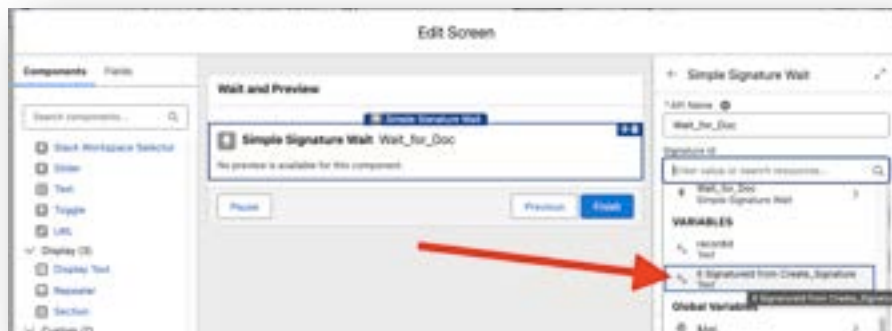


- 5) Next, we will create a signature record, already linked to the Lead record.
- Give it the **Label** and **API Name** you want
 - Select “Manually” in the **How to set record field values**
 - Select “Signature” as the **Object**
 - Click into the first **Field**, and select the “Lead” field, and for the **Value** choose the variable you created at the beginning of these steps
 - Click to **+ Add Field**, click into the Field and select the “Experience ID” field, and then hard-code a value that you will have gotten when you created the Signature Experience in the prior chapter. This Signature Experience will need to have a “templates” parameter set, which references the ID of the Google Doc to run.
 - Add more fields for the “First Name”, “Last Name”, and “Email” and in each case, click into the Value, and in each case, click into the “Get Records” step you created on the last page, to browse through the Lead fields until you find the matching Lead field.



SIMPLE SIGNATURE FOR SALESFORCE

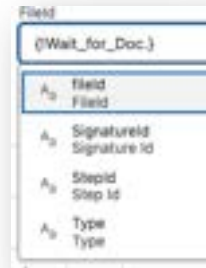
- g. Add “Attach PDF?” and “Embed PDF?” and give them each a TRUE Global Constant.
 - h. Add “Status” as a field and give it the value of **Requested**.
 - i. Add “Templates” as a field give it the value of the Google Doc template ID that you also added to the Signature Experience.
 - j. Add “Send Type” and select **None** from the dropdown.
- 6) Now we need to create a screen that will wait until the Google Doc has generated a preview. First press the + icon and select the “Screen” option again.
- a. Give it a **Label** and **API Name** of your choosing.
 - b. Drag in the “Simple Signature Wait” component from the **Components** menu on the left. Click on it and give it an API Name of your choice.
 - c. For the **Signature Id** parameter, click into it, and from the **Variables** select the record that you just created in the last step:
 - d. You can leave the **Step Id** blank for this flow, but it can be used for Multi-Step signatures.



- e. For the **Type**, enter: **Preview** – the other option you can use is: **Final**, but that is in case you want the plugin to wait until the final PDF generated, which will not be created until after the signature has been taken.
- f. Press the **Done** button to close this Screen.

- 7) We will now add an optional step, which is to display the PDF as a preview. It is optional, because we can also display the PDF on the signature form itself. But let's do it for this example. Press the + Icon again, and again, select "Screen".

- Drag in the "Simple Signature Preview" component from the **Components** menu on the left and click on it. Give it whatever **API Name** you would like.
- For the **FieldId** parameter, click into it, select the Wait step you created in the prior step, and it will expand to show you the parameters. Choose the **fieldId** one. And then press the **Done** button.



- 8) The next step is to have the user sign. Click the + Icon again and select a "Screen" option again. This time drag in the Simple Signature Form component and fill in the parameters:

- Label** and **API Name** as desired
- In the Screen Properties, open the **Configure Footer** section, and Uncheck the "Show Footer" option. For this step, we need to control it.
- Set the **Experience Id** as in prior steps
- Parent Id** will be the variable you created at the beginning
- For the **Other Parameters**, you do need to select the signature you created earlier, or else you will end up with two signature records. Click into this parameter, and select the Signature Id just like you did when setting up the Wait screen in Step 6. But this time, after you select the step, click into the parameter again and pre-pend "id=" to whatever value is shown. You will end up with something along these lines:

```
id={!something_id}
```

If you want to have the PDF display on top of the form, you need to use the showattachment parameter here in the Other Parameters, as well. The Value for this parameter is going to be the same **FieldId** value as what you would have selected in the prior (optional) step when you were previewing the PDF.

```
id={!something_id}&showattachmentid={!something_wait.fileId}
```

- Style is optional, but we give a recommended CSS in the prior section of this guide, when giving the first Flow example.
- 9) The final step, which is optional is to do a repeat of Steps 6 and 7 above, with the only difference being that for the **Type**, you would enter: **Final** in Step 6, instead of **Preview**. Now you are done.

- 10) Click the **Save** button, and then click the **Activate** button.
- 11) You cannot Run this here – it will only work if you are in Lightning, so open a new tab and go to Setup → Object Manager. Search for Lead and go into this object.
- 12) Click on Buttons, Links, and Actions and press the **New Action** button.
 - a. Select “Flow” from the **Action Type** dropdown.
 - b. Select the Flow you just created from the **Flow** dropdown
 - c. Enter a **Label** and **Name** that you would like, and press the **Save** button.
- 13) Click on **Page Layouts** and then open the layout which you use. From the **Mobile & Lightning Actions**, select the Flow you just created and drag it into the **Salesforce Mobile and Lightning Experience Actions** area. Save the layout.
- 14) Now, navigate to a Lead, and test it out. If you did everything correctly, it should work. Good luck!

Automated Parent Records Updates

When a Signature form is submitted, with all fields populated as required, and a proper signature captured, the Signature record is either created or else updated. In either case, records related to the Signature should often be updated. Usually it is the Parent record that needs to be updated. For example, an Opportunity might offer a signature, and once signed, the Stage of that Opportunity maybe should be updated to indicate progress.

This type of update is typically handled through a Flow, or for the more adventurous, through an Apex Trigger, which would simply monitor the Signature object, and when a new one, or updated one reaches the “Signed” status, certain actions can be taken on a record (or more than one) related to the Signature. However, in some cases, this simple function does not suffice because data is being captured and is not saved in the Signature record (it is not one of the fields on the Signature form) and does not exist elsewhere in Salesforce (or is not easily found). For these situations, we have a specific feature called Field Updates. Here is how it works:

1. A record is added into the **Field Updates** custom setting. This record would typically be added immediately before sending the user to the signature form. It might consist of data captured at a prior step in the process.
2. The user Signs the form, or Cancels out of it.
3. If the user successfully Signs the form, then the data in the **Field Updates** record is written to the appropriate record in Salesforce, usually the parent record to the Signature. If the user Cancels and does not Sign, then the temporary updates are deleted.

Code to write the custom setting record prior to launching the Signature Form can be found in the “Classes” folder, under the name: “FieldUpdates”.

GitHub.com Simple Signature Sample Code Repository

<https://github.com/ToAFinish/simplesignature>



Following is a table which explains what information should be populated in each of the fields in the **Field Updates** custom setting record:

Field Name	Populate With...
Name	Unique value, we typically recommend using a combination of the ParentID, the UserID, and a Numerical value.
signature__DefaultValueField__c	API Name for the formula field on the Signature Form to show this default value (only shows if the field is blank).
signature__Field__c	API name of the field in the parent object where a value will be written to.
signature__ForceOverwrite__c	Yes to overwrite value in existing Field, even if blank.
signature__RecordId__c	The ID of the parent record where data will be written to after signing.
signature__Status__c	By default the value "Ready" should be written here.
signature__UpdateType__c	Type of value to be written to this field. Options are: Text, Numeric, Date, DateTime, Boolean.
signature__UserId__c	The user ID of the signee.
signature__Value__c	Value to be written to the field in the parent object (if Text field).
signature__ValueBoolean__c	Value to be written to the field in the parent object (if checkbox field).
signature__ValueDate__c	Value to be written to the field in the parent object (if Date field).
signature__ValueDateTime__c	Value to be written to the field in the parent object (if Date/Time field).
signature__ValueNumeric__c	Value to be written to the field in the parent object (if Numeric field).

Please be aware that once the Signature Form is signed and the field is updated in the parent object, this custom setting record will be deleted.

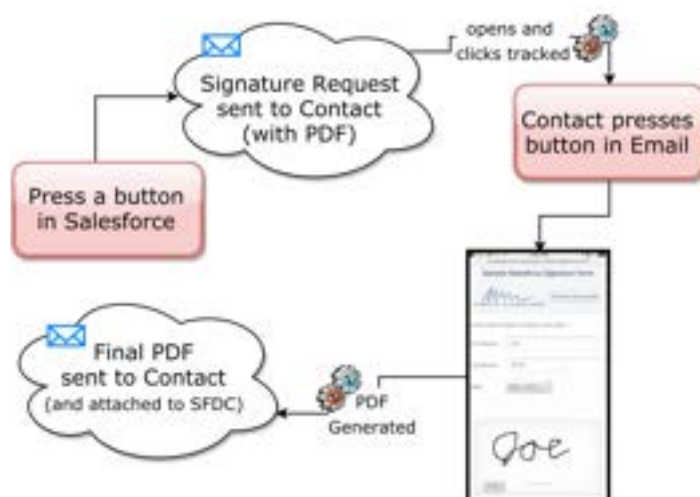
4 – Signature Requests (paid)

Steps necessary to set up Salesforce for requesting Signatures

The first two usages of Simple Signature, where you sign forms independently in the free version, and where you can set up Signatures to be tied to a parent object, are both fairly easy to set up and use. Requesting Signatures from a Contact or Lead in your Salesforce org is a bit more complicated, and requires several steps in order to work correctly:

- 1) Create a Google Doc template (or Visualforce page template) to be used in the Signature process. This will be the PDF that you will want to be the final result of the signature process. (Creating this template is covered in the next chapter.)
- 2) Configure the Salesforce Site to allow external people to sign your forms.
- 3) Optionally configure a Signature Experience to customize the signature process.
- 4) Create a way to trigger the request: button, flow, etc.

The end result of setting all of this up will be a full-cycle signature request process that can be used in one or more places at your organization.



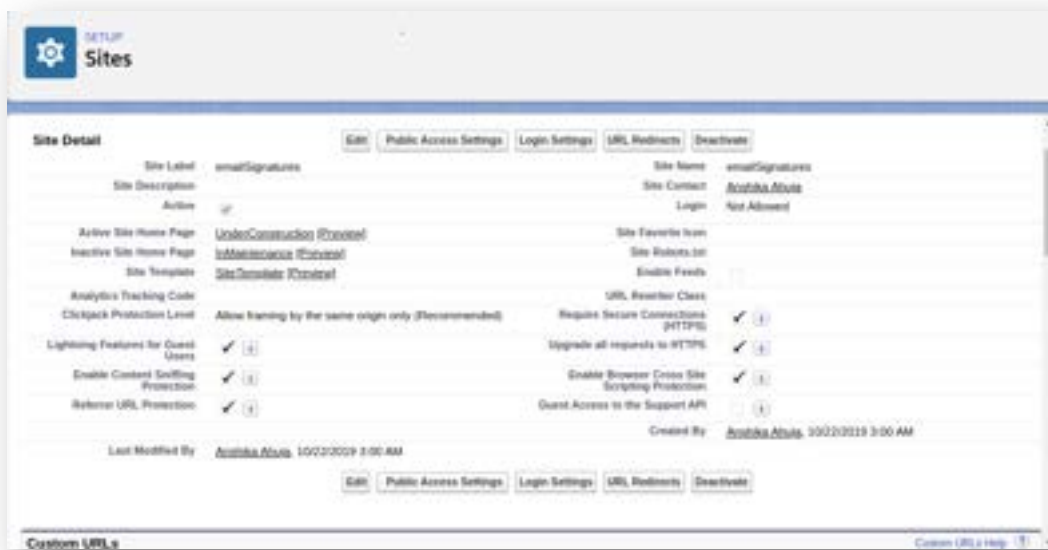
PDF Document Creation

Unless you wish to use the standard PDF that comes with our app, you will need to create a PDF document either using Visualforce or Google Docs. Please see the next chapter for this very important, and likely the most complicated step.

Site Setup

This part of the configuration will be complicated if you do not know Salesforce Sites well. Ask a Salesforce Administrator to help you set it up, if you haven't done it before. If you need us to help, we can do that. Email us at support@toafinish.com

- 1) Go to the Setup → User Interface → Sites and Domains → Sites and create a new Site. If you have not set up your Domain Name it will ask you to do this before you can create a site.



- 2) The site you set up can be called anything you like, and the critical items to consider are as follows:
 - a. For the **Active Site Home Page**, select the “SiteSignature” page that comes with Simple Signature, unless you want to use a custom page, which you can read about later in the chapter.
 - b. For the **URL Rewriter Class**, select the “signatureURLRewriter” class.
 - c. Deselect the **Cache public Visualforce pages** option on the Site.
 - d. Now **Save** the new Site in order to continue.

- e. You will need to know the **Site URL**. You will need this in the next step, so copy this to a text file for later use.
 - f. You will need to add these pages (at least) into the Site Visualforce Pages:
 - i. signature.SiteSignature
 - ii. signature.TrackEmail
 - g. Now, give this user needed rights: click on the **Public Access Settings** button, then on the **View Users** button, and click on the site user. For this user, assign them the “Simple Signature (Guest)” permission set.
 - h. This last step will give the signee access to fields like First Name, Last Name and Email that are in the Signature object. But if you create custom Signature fields in order to show, or accept other data from the signee, you will need to manually give access to the field(s) containing this data. For example, if you create a new field on the Signature object called “Secondary Email Address” and add it to the Field Set, when the recipient goes to sign, they would not see the field or be able to enter any value in it, until this is done. To give access to the field, go to **Public Access Settings** again (as in the step above) and then depending on your profile settings, give access to the field in a different way:
 - i. For the new Profile editor, click the **Object Settings** link towards the top of the Apps page, and then and scroll down and click on the **β Signatures** item. Next, press the Edit button again and give Read AND Edit Access to the fields you are including on the Signature Form.
 - ii. For the classic Profile editor, scroll down until you see the “Custom Field-Level Security” section and find the **β Signature** object and press the “View” link. Next, press the **Edit** button again and give Read AND Edit Access to the fields you are including on the Signature Form.
- 3) Next, we need to give access to the site to needed Signature records. Go to Setup → Security → Sharing Settings. For any object that your signees will access in the Signature process, please do the following steps:

The reason we need to do this is because the Site Guest User should no longer own records and they also cannot see any records that they do not own, effectively shutting them out of seeing any records at all.

- a. Scroll down to the **Sharing Rules** to the Signature object, and click on the **New** button on the related list.
 - b. Give it any rule name and label that you want.
 - c. Select “Guest user access, based on criteria” for the **Select your rule type**.
 - d. For **Select which records to be shared** set up some criteria that will give you access to either ALL records, or the records that will be signature requests. We recommend that you give access to all records where the Status <> “Signed”. All other statuses might be needed for access, but once Signed, the Site User no longer needs access to these records.
 - e. Select the user for the Site you will be using for signature requests under the **Select the users to share with**.
 - f. The **Access Level** is always going to be Read Only, so simply click the **Save** button and wait for a bit until the sharing rules finish running. Then, go back and repeat these steps for the Parent object, and any other object that you link to in the Signature Form.
- 4) Now, with your Site URL in hand, proceed to the next step. *Keep in mind that you might have to come back to this one in order to adjust the security of the site after you test following the next section.*

App Configuration

Once you have the Google Doc (or Visualforce page) that you want to use as the default for PDFs to be generated for signatures, and you have configured a Site to use it on, you will need to configure the app to use all of this together.

Ensure you are in the **Lightning Experience** (some of this will not work in Classic), and Go to Setup → Installed Packages → Simple Signature and click on **Configure**. If the paid version is enabled, you will see these sections in the Configuration tab:



- 1) Click the “Configuration” tab and paste the **Site URL** from the site you set up a few pages back into the “Site URL for Signature Requests” field.
- 2) Turn on the **Process Signature in Batch Mode** option.
- 3) Turn on the **Email Requested Signature with PDF Attachment** option, if it isn’t already.
- 4) **Save Settings** at the bottom of the page.
- 5) Click the “Site Configuration” tab and IF you see a message telling you to set up a Remote Site, then please do so.
- 6) Go back to the Setup section, if you are not already there, and search for “External Client App Manager”.
- 7) Click the **New External Client App** button.
- 8) In the “Basic Information Section”, provide **App Name**, **API Name**, and **Contact Email**. The rest can be left empty if desired.
- 9) In the “API (Enable Oath Settings)” section, turn ON **Enable OAuth**.

Setting Up an External Client App

The following instructions are somewhat technical, and could be in flux for some time. Salesforce recently introduced External Client Apps as a replacement for Connected Apps, and they are a bit trickier to use in this environment.

An “**INSUFFICIENT PRIVILEGES**” error on one of the upcoming steps (usually step #15) means you will need to open a case with Salesforce tech support so that they enable “External Client Apps” in your org.

- 10) In the “App Settings” section, provide the callback URL for login or sandbox. Use:
 - a. **Production:**
`https://login.salesforce.com/apex/signature__SiteAccessAuthorize`
 - b. **Sandbox:**
`https://test.salesforce.com/apex/signature__SiteAccessAuthorize`
- 11) Add only these required **OAuth Scopes**:
 - a. Manage user data via APIs (api)
 - b. Manage user data via Web browsers (web)
 - c. Perform requests at any time (refresh_token, offline_access)
- 12) Under “Security”, **uncheck** the “Require Proof Key for Code Exchange (PKCE) extension for Supported Authorization Flows” option.
- 13) Click **Create** to create the app.
- 14) Go back to External Client App Manager and open the created app and go to the **Settings** tab
- 15) Under “Auth Settings”, click on the **Consumer Key and Secret** button. It will go through secondary verification to ensure you can see these codes.
- 16) Copy the **Consumer Key** and **Consumer Secret** into a notepad for later.
- 17) Still in Salesforce Setup, search for “Installed Packages” and find the “Simple Signature” app. Click on the **Configure** link for this app.
- 18) Click on the **Site Configuration** tab.
- 19) Under “External Client App Settings”, enter the Consumer Key in the **Client ID** field and the Consumer Secret in the **Client Secret** field, which were saved from earlier.
- 20) Then click the **Save Settings** button.
- 21) Now, click the **Authorize** button below. *(For this you might need to switch to CLASSIC mode, if it gives you an error here.)* Accept the permissions and log into Salesforce using the login of yourself or another user that is specifically given all rights to the Signature and other objects that are needed for the Signature process. This user must also have one of the Simple Signature permission sets. Once you log in, after a couple seconds, above the button, it should show the date and user who authorized, signifying that you completed it correctly. If not, close out and open the Configure page again (in Classic mode), and confirming that the Consumer Key and Secret are saved, press the "Authorize" button again.

SIMPLE SIGNATURE FOR SALESFORCE

On the following page are listed all of the other options related to Signature Requests on the Configuration tab. Read over them to see if there is anything you need to configure in order to meet your requirements.

Option	Description
Create Lead on Save	Creates a Lead linked to the Signed when the form is signed. Turned off by default, as this can now easily be done via a Flow.
Run PDF Automation on Save	Generates the PDF when the form is signed.
Attach PDF After Signature	Attaches the PDF generated to the Signature record.
Attach PDF on Parent Record After signature	Attaches the PDF to the parent of the Signature record.
Attach PDF on Contact Record If Not Parent	If a Contact is linked to the Signature record, but it is not the parent object, this will also put the PDF on the Contact.
Email PDF to Contact After Signature	Emails the generated PDF to the Signee.

** Please note that the three Attach options and the Email option can be overwritten for any particular Signature.*

Option	Description
Template Visualforce Page for PDF	This is the Visualforce Page to be used by default (if no Google Docs page or other Visualforce Page are set for the particular signature.) <i>Please note that if you use a custom Visualforce page, you must prepend "c__" to the name. So, a VF page called HelloPDF would be c__HelloPDF.</i>

SIMPLE SIGNATURE FOR SALESFORCE

Option	Description
Maximum width (in pixels) for signature field	This will set the width of the Signature in the “Signature” field (a rich-text field) on the Signature record. Only Width OR Height should be used.
Maximum height	This will set the height of the Signature in the “Signature” field (a rich-text field) on the Signature record. Only Width OR Height should be used.
Site URL for Signature Requests	This is the URL for the Site that is created for Signature Requests. Make sure to include the final slash (/).
Use Files instead of Attachments (<i>on by Default</i>)	Will put the Signature images and the PDF documents in the Files related list instead of the Notes & Attachments list. <i>Please note that if this is checked Salesforce will also show the files to the Attachments, so it will appear to be in both lists.</i>

Option	Description
Turn off Emails	When this option is checked, NO emails are sent out, other than the specific email addresses or domains added to the Whitelist. Whitelisted emails should still receive the emails. When this option is UnChecked all email should go out as usual, Except the email addresses or domains added to the Blacklist.
Blacklist and Whitelist Emails/Domains	Add the Emails or domain in either the Whitelist or Blacklist
Process Signatures in Batch Mode (<i>On by Default</i>)	This option is strongly recommended. It turns on the processing of Signatures in “batch” or asynchronous mode. This means that no matter how the Signature record is created, it will still be processed.
Email Requested Signature with PDF Attachment (<i>On by Default</i>)	If you are setting up signature requests, this option is recommended as it will attach the PDF generated to the email. If you do not turn this on, then you should turn on the “Send Email” workflow rules included in the installation.

SIMPLE SIGNATURE FOR SALESFORCE

Option	Description
Default Signature Request Send Email (Uses Org-Wide Email)	This field can be used to enter one email address that should be used as the “From” email when sending a signature request and when sending the final, signed PDF. Please note: 1) This email must be in the Organization-Wide Email Addresses and must be validated. 2) If this is not used, then the currently logged in user will send the Signature Requests and the default Site authorization user (see page 39) will be the sender of the final PDF.
CC Record Owner When Emailing Signed PDF	Will copy the Salesforce user who created the Signature Request when the final, signed PDF is sent to the Signee.
CC Record Owner’s Role Hierarchy When Emailing Signed PDF	Will copy the manager (and manager’s manager, etc.) of the Salesforce user who created the Signature Request when the final, signed PDF is sent to the Signee.
CC Other Email Address When Emailing Signed PDF	Will copy a hard-coded email address when the final, signed PDF is sent to the Signee.
BCC Record Owner’s Email to Salesforce When Emailing Signed PDF	If Email to Salesforce is turned on for the user who initiates the Signature Request, then this unique email to Salesforce address will be blind copied on the email that goes to the Signee after they sign. This will have the effect of logging the email into History.

Option	Description
Send Email to Partner	This should be turned on only if you are working with To A Finish technical support to enable the Partner option, where the signature requests are hosted in our Org. This is used in particular with Profession edition.
Partner Email Address	This email address will be given to you by To A Finish technical support.

Turn on or off the options according to what is defined below and fill out any fields that are necessary. Then click the **Save Settings** button.

Signature Requests Custom Pages

Some companies will be able to use the default signature page that comes with the app, but if you find yourself in need of 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 at our hourly rates.

- 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 blank 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 "Custom_SiteSignature" 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!

- 4) Next, 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) Finally, you need to set this page as the one to be used by your site. Go to Setup → User Interface → Sites and Domains → Sites and then Edit your signature site. In the **Active Site Home Page** field, type the name of your new Visualforce page, or use the search feature to find it. Now **Save** the site and you are done.

Let us know if you need help: support@toafinish.com.

Launching the Signature Request

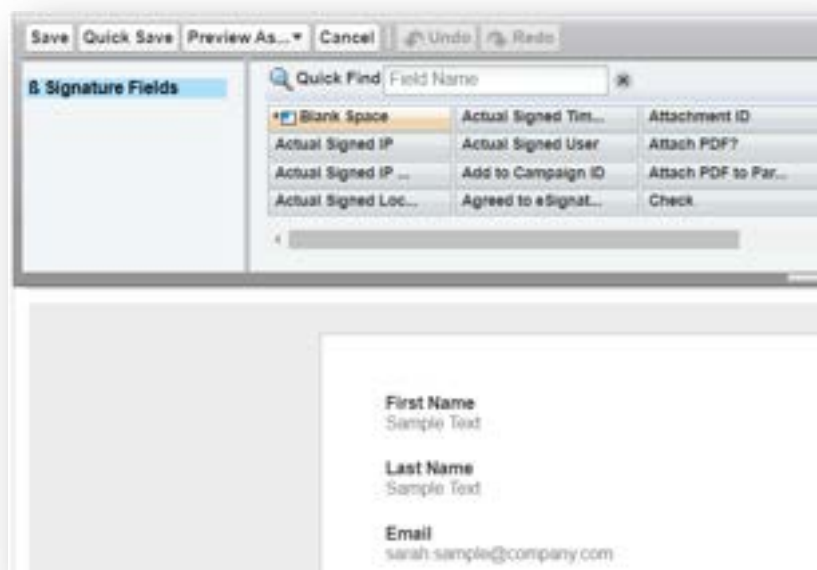
The next thing to do is to send the Signature Request. We have many ways of doing this, but let's start with a manual option. Let's set up a button on the parent object of the needed signature that will allow you to request it. For example, if you need a signature on a Case, or an Opportunity or on your own custom object, you would set up the button on this object. We will use Leads as the parent object in the examples below.

You can create Signature Requests via any other method which can create a Signature record. Here are a few standard options we recommend:

Actions

For Lightning and Mobile use, the best way to create Signature Requests is via an Action. Actions are fast and they can be used in offline mode, if configured correctly.

- 1) Make sure you are in the "Lightning" web version of Salesforce, with Admin permissions.
- 2) At the top-right of the page click on Setup → Object Manager → Lead → Buttons, Links and Actions, and click the **New Action** button.
- 3) Select "Create a Record" as the **Action Type**, select "Signature" as the **Target Object**, Give it a Name and Label like "Request Signature", and finally set **Success Message** to something like "Signature Request Created".
- 4) In the Layout, add fields such as First Name, Last Name and Email (our minimum recommended fields) in the layout for Signature Request. These are the fields you will see and be able to modify when creating the request. **Save**.



- 5) After you save the Action, you will see a list of “Predefined Field Values.” Press the **New** button to add default values for fields. These are the minimum recommended fields, and they use the Lead object as the example:

Field Name	API Name	Field Type	Value
Attach PDF to Parent?	signature__AttachPDFtoParent__c	Checkbox	TRUE
Attach PDF?	signature__AttachPDF__c	Checkbox	TRUE
Email	signature__Email__c	Email	Lead.Email
First Name	signature__Name__c	Text	Lead.FirstName
Last Name	signature__Name_2__c	Text	Lead.LastName
Status	signature__Status__c	Picklist	Requested

** Other recommended fields are EmailPDF?, Embed PDF?, Requested Form, Requested PDF Form, Lead or Contact (if neither of these are the Parent object) and Templates. For multi-step signature processes, you would also use the MultiStep field set to TRUE.*

- 6) Go to Setup → Object Manager → Lead → Page Layouts → Click on Mobile & Lightning Action and add it into the page layouts to allow a user to click on it to request a signature.

Lightning Components

Another option for creating Signature Requests in Lightning and Mobile use is to create a Lightning component the same way that is described in chapter 2. A few things to note:

- 1) Lightning Components do not support offline mode, so this method will not always work in Salesforce Mobile.
- 2) These parameters from chapter 2 are used exclusively for signature requests:
 - a. **Allowrequest** turned on enables the requesting of signatures,
 - b. **embedpdf** will embed the PDF to be signed immediately above the signature form - enabling the signee to review it,
 - c. **generateqrcode** will generate a QR Code image that will be generated at the time of the signature request and will display the image of the QR Code in the popup window, rather than closing the window. This allows the person requesting the signature to ask a signee to scan the QR Code in order to be sent to the signature page, and
 - d. **multistep** will generate a Step record in addition to the Signature record and relies on the admin to set up additional Step records triggered off of the first one.

Email Template Setup

The final step you will need to take is to configure the two Email Templates that go out as a part of the Signature Request process. First, we will work on the email that is sent out with a signature request:

- 1) Go to Setup → Installed Packages → Simple Signature → Configure and click on the “Email Templates” tab.
- 2) Click **Edit** next to the “Signature Request” entry, and then expand the “HTML Body” section.
- 3) Update the Subject line to what you want the recipient to first see.
- 4) Use the WYSIWYG template editor to modify the email, adding wording or even pictures. However, please be careful around the “Sign Online” button, as it contains hidden values. You should likely press the code button shown here:  in order to edit the actual code of the HTML email template.

** Please note that an unsigned version of the PDF you are requesting them to sign will be attached to the email, so you might want to say something about it.*

- 5) When you have it just right, also remember to edit the “Text Body” section for those recipients who have text-only email readers.

Next, you will want to edit the email sent after the signature process finishes.

- 6) In the same screen, on the row with the "SignatureReturn" template, click the **Go to Edit** link. This will take you to the email template, if you cannot edit it:
 - a. Click the “Setup” link in the body of the email, and this will take you to a classic screen that will allow you to edit.
- 7) Once in the email, click on the "Edit HTML Version" button.
- 8) Update the Subject Line to say what you want it to say, and then update the text of the email.
- 9) Save the email.

Testing

You are now ready to test the full-cycle functionality. Go to a record of the type you determined to be the parent of the Signature object and press the "Request Signature" button. After a few seconds, the contact or lead connected to that parent object should receive an email request. When they fill it out, they should then receive the completed PDF, with the embedded signature.

If you run into any issues, please shoot us an email at support@toafinish.com. We will be happy to help.

Advanced Features

Following are a few more advanced features which can be used in the Signature Request process. They are not given step by step instructions as they should be performed by a Salesforce Administrator with experience in this area. If you run into issues, please reach out to us and we will be able to help.

Send From Emails

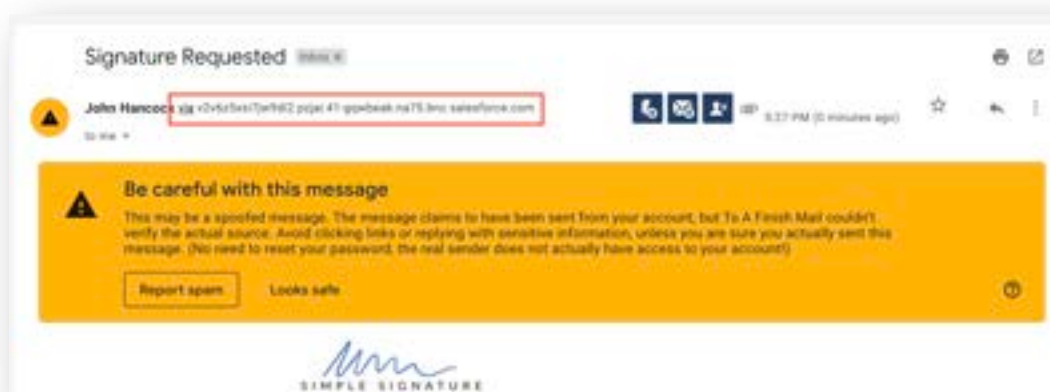
By default, signature Requests will be sent out from the user who creates the request. This is usually the desired result.

However, when the signer completes the signature request, the signature is processed by the Authorized user that has been set up on the Site, so the email will be sent to the signer from this user account. This is not necessarily the best experience. There are two ways to get around this:

- 1) In the Configure page, you can enter an email address in the “Default Signature Request Send Email” field. If you do, both the Signature Request, and the Signature Confirmation (with the final PDF) will come from this email address. However, you **MUST** create and verify an Organization-Wide Email Address that matches the email you enter in this field in order for this to work.
- 2) Alternately, you can create Organization-Wide Email Addresses for all of your users. If you do this, then both the Signature Request and the Signature Completion emails will come from the user who created the signature request. However, if an email address is not found for user in the Organization-Wide Email Addresses, then it will look for the default user in the Configure page, and if that one does not have an Organization Wide Email Address, then it will be sent from the Authorized user.

Email SPAM and Other Problems

When Salesforce sends emails from users in the system, it will look like the emails are potentially SPAM, or worse, unless careful configuration is followed. For example, look at the following Signature Request email:



Notice that, not only is there a big “Be careful with this message” warning, but there is also an indicator next to the name of the Sender which indicates that this email was sent through an unrecognized process.

These issues can be resolved via careful configuration of both the Salesforce system and the email or internet domain. If you have a Salesforce Partner who can do this for you, please ask them to configure your Salesforce system so that it is fully authorized to send email on your behalf. If you do not have a Salesforce Partner who can help you, we do offer services to do this as well. If so, reach out to us at support@toafinish.com.

Multiple Step Signatures

Simple Signature is most often used for one individual to sign a document or agreement. However, with a little configuration it can be used to set up a complex, multi-step Signature process using the Step object and a Flow.

- 1) To get started with Multiple Steps, please first create the Signature Request as explained on in the “Launching the Signature Request” section earlier in the chapter. The only change is that the **MultiStep** option needs to be turned on.
 - a. For Signature Requests created by an Action or an automated method, simply populate the `signature__MultipleSteps__c` to TRUE.
 - b. For Signature Requests created using an Experience ID, the “multistep” parameter must be checked.

- 2) When this “MultiStep” option is on, there are two significant differences that happen immediately:
 - a. A new “Step” record, the child of the Signature record will be automatically created when the Signature record is created. This new record will be the foundation of the signature process, so that only when all Step records are signed, will the signature process be finalized.
 - b. The email template that is sent for Signature Requests with multiple steps is different from the regular one. By default, the “ESignMutliContact” Classic Email Template is used for Signature Requests with multiple steps, instead of the regular “SignatureRequest” one.
- 3) Adding additional required signatures is easy. Simply create new Step records when the first one is created. You can use a Flow to do it, based on the Step record being created with the “Order” field having a “1” value. The Step record with the Order = 1 field will be the first step, which is created automatically. Other fields to populate in the new step records are:
 - a. First Name = The first name of the signee. Often used in the email template.
 - b. Last Name = The last name of the signee
 - c. Email = The email address the Signature Request should be sent to
 - d. Order = 2, 3, 4, etc. (numerical value, setting the order you want the signature requests to be emailed)
 - e. Email Template = Unique name of the Email Template (if different from the ESignMultiContact default template). *The Email Template (completion) parameter needs to also be set in the Signature Experience itself.*
 - f. Status = “Requested”
 - g. Wait On Prior Step (Set to TRUE so that this signature request is only emailed once the prior one - order is defined by the Order field - is Signed. Set to FALSE if you want it to be sent at the same time as the prior one.)

Once you have created the new step and populated the necessary fields, it should look something like the image on the next page, if you are using a Flow to create it.

Create a Record of This Object

* Object
8 Step

Set Field Values for the 8 Step

Field	Value
Email Template (Request) X	ESignMultiContact
Email X	priyankprakash123.1696@gmail.com
First Name X	SecondSigneeFName
Last Name X	SecondSigneeLName
Order X	2
Signature X	Triggering signature_Step_c > Signature X
Status X	Requested
Wait On Prior Step X	True X

+ Add Field

☐ Manually assign variables

Check for Matching Records ☐

- 4) Once you have set up the Flow to create the needed Step records, save it and Activate it. It should launch whenever the Signature record is created.
- 5) Note that when all Steps in the Signature Process have been signed, the Signature record Status field will be set to “Signed” as well, which triggers the final PDF being generated. The remaining task is to bring the Signature information from the Steps into the main Signature object so that we can display them in the PDF. Now, this step is not mandatory, as you may not need all signatures on the final PDF, or if your PDF is generated using Visualforce, you might be able to build an Apex class that pulls the signature images in from the Step records themselves. For those not so lucky, here are the steps to do either in a Flow:

SIMPLE SIGNATURE FOR SALESFORCE

- a. For each Step, create a Signature Image Rich Text field in the Signature object. You might want to also create a Signature Date and Signature Name (signed by) for each Step as well.
- b. Create a Flow triggered on the Step object that will copy the following fields, based on the Order field. The automated Step that is created with an Order value of 1, will copy the Date and Signature image to the Signature record, but you must do it for any Step with an Order value of 2 or 3 or greater. Here is what the filter of the flow could look like:

The screenshot shows the 'Configure Start' section of a Salesforce Flow. It includes the following configuration:

- Object:** Step
- Configure Trigger:**
 - Trigger the Flow When:
 - ☐ A record is created
 - ☒ A record is updated
 - ☐ A record is created or updated
 - ☐ A record is deleted
- Set Entry Conditions:**

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements: All Conditions Are Met (AND)

Field	Operator	Value
signature__Status__c	Equals	Signed
signature__Order__c	Equals	1
signature__SignatureImage__c	Is Null	False

+ Add Condition

- c. Here are the fields to copy (the field names are just examples) in the case where signature__Order__c = 2 and signature__Status__c = "Signed" and on the next page an image shows what it might look like:

Target Object	Target Field	Source Object	Source Fields
Signature	SignatureImage2__c	Step	signature__SignatureImage__c
Signature	SignatureDate2__c	Step	signature__Signed__c
Signature	SignatureName2__c	Step	signature__Firstname__c & " " & signature__LastName__c

SIMPLE SIGNATURE FOR SALESFORCE

The screenshot shows the configuration for a Salesforce Workflow Rule. The rule is named 'Signature' and is set to update records of the 'Signature' object type. The filter condition is 'All Conditions Are Met (AND)' with the field 'Id' equal to '\$Record > Signature'. The rule sets three field values: 'SignatureEmail2_c' to '\$Record > Email', 'SignatureImage2_c' to '\$Record > Signature Image', and 'Signature_Date_2_c' to '\$Record > Signed'.

Update Records of This Object Type

* Object
Signature

Filter Signature Records

Condition Requirements to Update Records
All Conditions Are Met (AND)

Field: Id Operator: Equals Value: \$Record > Signature

+ Add Condition

Set Field Values for the Signature Records

Field: SignatureEmail2_c Value: \$Record > Email

Field: SignatureImage2_c Value: \$Record > Signature Image

Field: Signature_Date_2_c Value: \$Record > Signed

+ Add Field

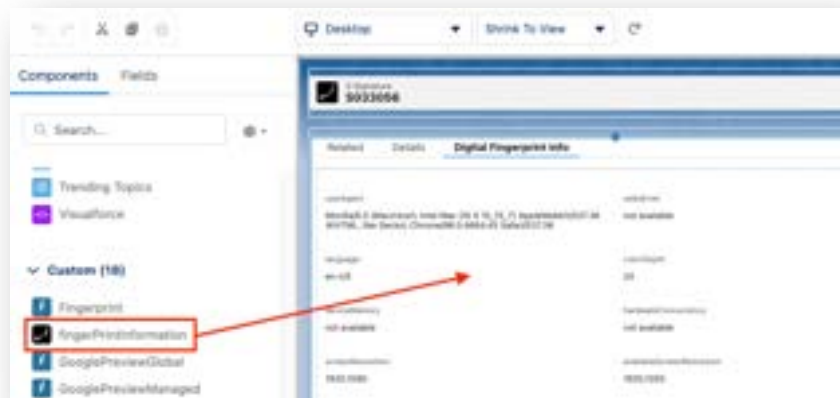
- 6) That is all that needs to be done to implement a robust multiple signature solution. You are ready to test it out!

Signature Site and Data Security

Electronic Signatures have a degree of potential questioning that hard-copy, paper document signatures do not have. How do you determine if the person you intend to sign is actually the person who signed? Well, with Simple Signature you have a few things that will help:

- 1) You can set a password that must be typed in before the signee can see the signature form. You can do this by simply filling in the **Signature Password** field using a flow or trigger. This will ensure that only the signer will be able to sign, even if someone unauthorized has access to the emailed signature request.
- 2) You can set an expiration for the signature, so that if the intended signee does not sign within a set number of days, then the signature request is no longer valid. You do this by populating the **Expiration Date** field with a date in the future.
- 3) During the Signature process the **Digital Fingerprint** field is populated with a 32-character string, which is supposed to be unique for each device. In this way, you can determine if signatures are signed on the same device or not.

If you want to see exactly what information the digital fingerprint value contains, you can add the **fingerPrintInformation** Lightning component to your Lightning page for the Signature object.



- 4) The IP Address of the signee is saved in the **Actual Signed IP** field, and a secondary field called **Actual Signed IP Info** attempts to collect information about the location of the IP address of the signee. If the Info field does not work, you can also create a formula field that contains a link to the IP address information. Use code similar to this to create the formula:

```
if ( not ( isblank ( signature__IPWhenSigned__c ) ), signature__IPWhenSigned__c & " " &
HYPERLINK("https://ipapi.co/"+signature__IPWhenSigned__c+"/json/", "info","_blank"), "" )
```

SIMPLE SIGNATURE FOR SALESFORCE

- 5) The **Actual Signed Location** field will contain the Latitude and Longitude of the signee at the time of signature. This data is dependent upon the signee's device providing the location. You can also create a formula field with a link to map this location. Use code similar to this in the formula:

```
if ( not ( isblank ( signature__ActualSignedLocation__Latitude__s ) ),
text( signature__ActualSignedLocation__Latitude__s )+" "+text(
signature__ActualSignedLocation__Longitude__s ) + " " +
HYPERLINK("http://www.google.com/maps/place/"+text(
signature__ActualSignedLocation__Latitude__s )+", "+text(
signature__ActualSignedLocation__Longitude__s ), "map", "_blank"), "" )
```

- 6) One more formula which might be of use when determining if a signature is valid can be seen below. It calculates the number of Days, Hours and Minutes between when the Signature Request was emailed for signature, and when the Signature was actually completed. A blank or problematic value indicates a need for further review:

```
if( if( isblank ( signature__ActualSignedTimestamp__c ), Now(),
signature__ActualSignedTimestamp__c ) - if ( isblank ( signature__Emailed__c ),
CreatedDate, signature__Emailed__c ) > 0 , "D:"

& TEXT( FLOOR( if ( isblank ( signature__ActualSignedTimestamp__c ), Now(),
signature__ActualSignedTimestamp__c ) - if ( isblank ( signature__Emailed__c ),
CreatedDate, signature__Emailed__c ) ) ) & " H:"

& RIGHT( "0" & TEXT( FLOOR( MOD( ( if ( isblank ( signature__ActualSignedTimestamp__c ),
Now(), signature__ActualSignedTimestamp__c ) - if ( isblank ( signature__Emailed__c ),
CreatedDate, signature__Emailed__c ) ) * 24, 24 ) ) ,2) & " M:"

& RIGHT( "0" & TEXT( ROUND( MOD( ( if ( isblank ( signature__ActualSignedTimestamp__c ),
Now(), signature__ActualSignedTimestamp__c ) - if ( isblank ( signature__Emailed__c ),
CreatedDate, signature__Emailed__c ) ) * 24 * 60, 60 ), 0 ) ) ,2), "" )
```

** This formula can also be tweaked to only pull the number of days, or hours, for example, so that you can then set a limit on signing if more than a predefined period of time has elapsed since the emailed request for signature.*

All of these measures are useful for ensuring the correct person signs the requested document, though they cannot ever guarantee it.

Signature URL Security

When sending a signature request, you are including a link to the Salesforce Site, which gives away the Id for the Signature Request record. If everything in your site is configured properly, this should not be a problem. The only potential issue could be if a recipient of a signature request were to manually change the Id on the URL in a sequential manner. This would potentially allow them to see a signature request that was intended for another signee.

This possibility will not be a problem for some organizations because they are not providing sensitive information in the signing area, but if it is potentially an issue, then we have an advanced URL-shielding mechanism which is explained below. When implemented, instead of the record Id, the end-user sees a random 32-character string, which is not sequential and so would be virtually impossible to guess.

Here are the steps to set up this feature.

- 1) Go to Setup → Installed Packages → Simple Signature → Configure and then click on the **Email Templates** tab.
- 2) You will want to create a new Email template. But before you do, select **View** on the email template you are currently using. If you have not created an email template before, it would be the "EmailSignature" entry.
- 3) Select the entire body of the email, including the button in the middle and copy it to the clipboard. You will paste it later into a new template. **Close**.
- 4) Now click the **New** button. Fill out the Label, Name, Description (optionally), Subject, and check the **Active** checkbox. Save the "Name" value for later. You will need this in the last step of the process.
- 5) Now, expand the "HTML Body" section and paste the email you copied into the text area. **Save**.
- 6) Click on the **Edit** action next to the new email template you created and then expand the "HTML Body" section again. Now click on the </> button in the toolbar of the editor. This will switch you into code mode.
- 7) Scroll down in the code until you see the following:

```
data-cke-saved-
href="{!signature__Signature__c.signature__SiteURL__c}?id={!signature__Signature__c.Id}"
```

And almost immediately following, you will see something like this: ()

```
href="https://some_url_here/{!signature__Signature__c.signature__SiteURL__c}?id={!signature__Signature__c.Id}"
```

SIMPLE SIGNATURE FOR SALESFORCE

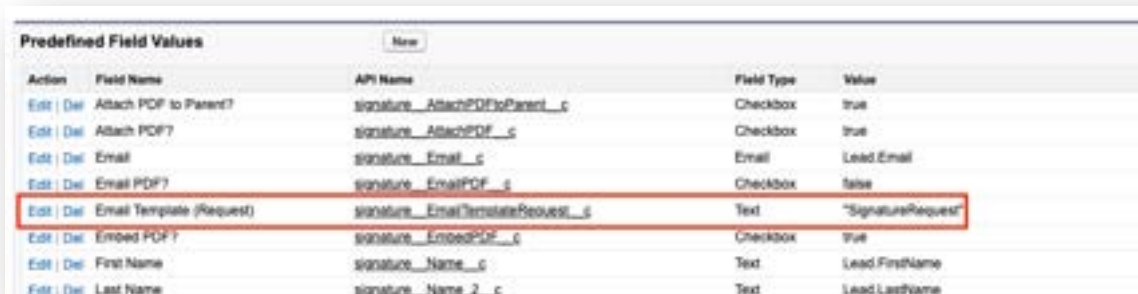
- 8) You want to replace the first one with:

and the next area, while removing any hard-coded values for the URL:

```
data-cke-saved-  
href="{!signature__Signature__c.signature__SiteURL__c}/{!signature__Signature__c.signature__  
_Signature_Path__c}"
```

```
href="{!signature__Signature__c.signature__SiteURL__c}/{!signature__Signature__c.signature__  
_Signature_Path__c}"
```

- 9) Next, you will need to add a **URL handler** to the Site. For this step, go to Setup → User Interface → Sites and Domains → Sites and Edit the Site you are using for Simple Signature.
- 10) In the **URL Rewriter Class** field, select "signatureURLRewriter" and **Save** the Site. Now the Site is ready for the new process.
- 11) Finally, you need to set your Signature Request to use the new Email Template you created. For this step, it will depend on whether you are using an Action to generate the signature request, or some other method, like a Lightning component.
- 12) If you are using an **Action**, simply take the "Name" of the Email Template and open the Action you are using to create the signature request. Click the **New Predefined Value** button and select the "Email Template (Request)" field name. Next, enter the name of the Email Template you created as the value for this field, enclosed in quotes. The end result will look something like this:



Action	Field Name	API Name	Field Type	Value
Edit Del	Attach PDF to Parent?	signature__AttachPDFtoParent__c	Checkbox	true
Edit Del	Attach PDF?	signature__AttachPDF__c	Checkbox	true
Edit Del	Email	signature__Email__c	Email	Lead Email
Edit Del	Email PDF?	signature__EmailPDF__c	Checkbox	false
Edit Del	Email Template (Request)	signature__EmailTemplateRequest__c	Text	"SignatureRequest"
Edit Del	Embed PDF?	signature__EmbedPDF__c	Checkbox	true
Edit Del	First Name	signature__Name__c	Text	Lead.FirstName
Edit Del	Last Name	signature__Name_2__c	Text	Lead.LastName

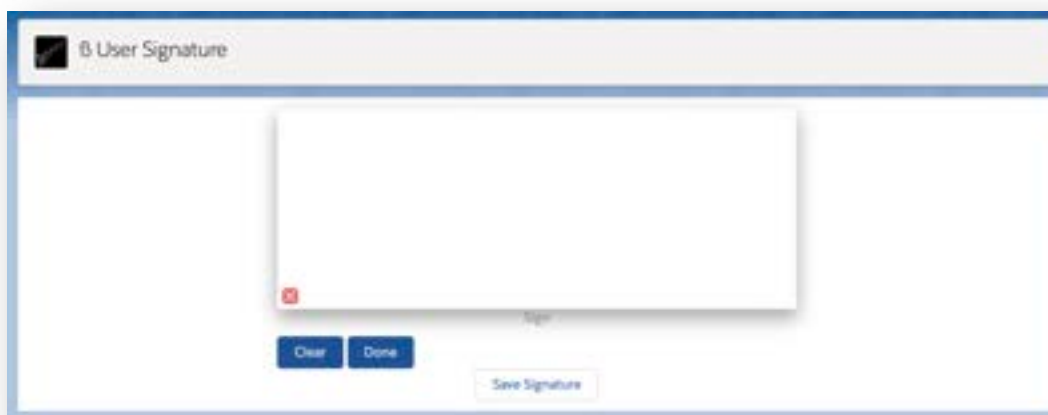
- 13) If you are using a different method to create the Signature Requests, you might have to use the **emailtemplaterequest** parameter in your Signature Experience.

And you are ready to test!

User Countersignature

Simple Signature allows users to store a signature to be used as a countersignature, which is applied automatically after the signee completes a signature process. In order to set this up, you first need to configure a few things.

- 1) Make sure you are in Lightning with Admin rights. Click on the **Search Apps and Items** icon on the top-left of the page, and type in Files in the search box. Click on **Files** when it comes up. 
- 2) Click on the **Libraries** link on the left list of folders, and once you see the Libraries, click the **New Library** button (assuming Simple Signature is not already in your library.). Enter "Simple Signature" as the name and **Save**.
- 3) Click into this new Simple Signature library and press the **Manage Members** button. Using this popup, add as many users to this library as may need to have counter signatures. If it is a lot of users, you might want to consider adding them from a public group.
- 4) Once all are added, each user who wants a Counter Signature should go to the Search Apps and Items icon and this time search for **User Signature**.



- 5) After each user who needs a counter signature has signed, then you will need to add the counter signature code to the Visualforce PDF or the Google Doc. For the Google Doc, see the example in the Merge Codes for Google Docs in the next chapter.

** Please note that the "Access Libraries" right is needed for Community users to use the "User Signature" functionality. This right is typically not on by default, so you will need to set it up before Community users can use this feature.*

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

5 – Document Generation (paid)

How to set up Simple Signature to work with custom documents, either through Google Docs, Visualforce code, or an external app...

Simple Signature is intended to be, well, simple. The toughest part might be the creation of the PDF documents where the final signature is supposed to go. For most companies, the included **Google Docs** integration is the simplest and fastest way to build these documents on their own. And for those geeks who love to code, creating Visualforce pages might be simple, and even fun. If neither of those options works, we have three other options:

- ❖ To A Finish, the creator of Simple Signature, offers a service to **create Google Docs or Visualforce documents** for your organization. Depending on how complicated your document is, they might typically cost a few hundred dollars to build. Contact sales@toafinish.com to explore this option in more detail. These documents can look exactly like you want them, and once they are built, there is no cost to use them.

Additionally, you also have several apps that work with Simple Signature and provide document generation (though we strongly prefer using Google Docs). These are listed at the end of the chapter, and will require significant technical work to get working. If you reach out to support@toafinish.com for help, we should be able to help.

Google Docs PDF Document Creation

Simple Signature comes with a very powerful document generation engine which employs Google Docs, an online word processor accessible to almost everyone. It will even work with the free version of Google (not the free version of our app).

To create a new Template in Google Docs, first you will need to configure Simple Signature in order to connect to your Google account:

- 1) Switch to **CLASSIC** mode. You will need to be in Classic mode to do these next steps. Once in Classic, go to Setup → Installed Packages → Simple Signature → Configure.
 - a. Click the “Google Doc Configuration” tab and IF you see a message telling you to set up several Remote Sites, then please do so.

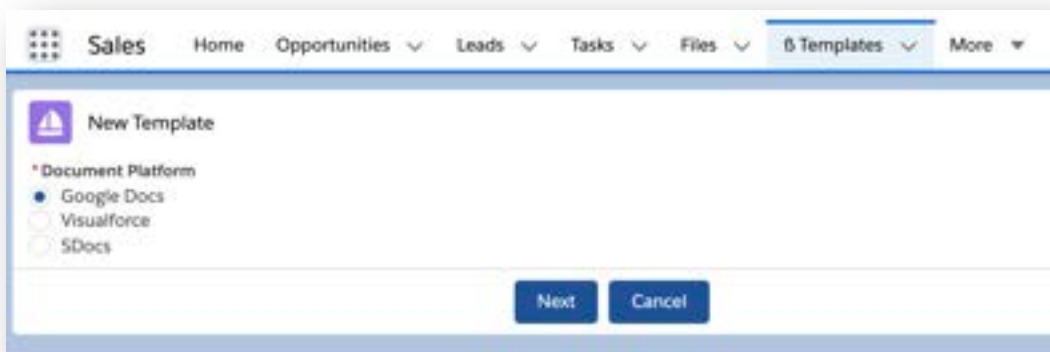
SIMPLE SIGNATURE FOR SALESFORCE

- b. In the Template Configuration sub tab, Click on Google APP dropdown and check “Native” and uncheck “Web Configuration” (This should be done by default)
- c. Click the **Authorize/Authenticate** button. (For this you must be in CLASSIC or it will give you an error here.) Accept the permissions and log into your selected Google account. Once you log in, it will give you a code to past into the tab, and after doing this, it should show you a **Revoke** button, signifying that you completed it correctly.

Please Note that a Platform User will not be able to authenticate. For some unexplained technical reason, it must be a Standard Salesforce licensed user.

After setting up Google Docs, you can now create as many Templates to be used with Simple Signature as needed.

- 2) Switch back to **Lightning** mode. Click on the **Search apps and items** icon, and search for Templates. Click on **Templates** and then you will see the option to create a New template.



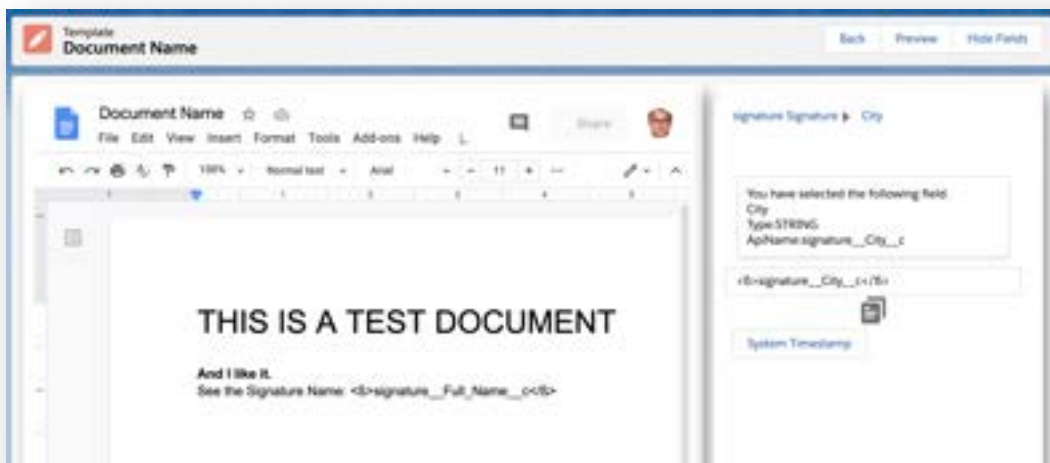
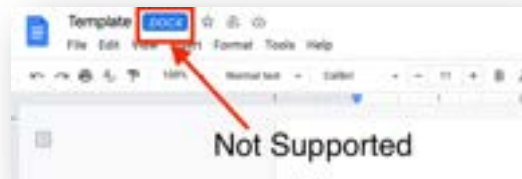
- 3) Select **Google Docs**, and press Next. As this is your first template, select **Create New Template** and press Next.
- 4) On the New Template page, enter the fields as shown below and press the **Save** button.



SIMPLE SIGNATURE FOR SALESFORCE

- a. Enter anything you want in the **Name** field. The Name is not used for anything programmatic, so you can truly call it what you want.
 - b. If you have an existing Google Docs document that you want to use as the starting point for this one, then paste in the **Google Id** for the document (taken from the URL when editing it) and click the Copy Template checkbox. For your first one, just leave these fields blank.
 - c. In the **Object Name** field, select “Signature” as the object. If you want to create other types of documents that are *not* intended to be signed, you can select a different object.
 - d. In the first **Permission Type** field choose “Anyone with link” to allow other users in your Salesforce instance to use this document as a Template for Simple Signature. The other options provide more specific access: User, Anyone in this group, and Anyone in this Domain.
 - e. In the second **Permission Type** field, choose “writer” so that other Salesforce users can edit this document. If you do not want anyone to edit except for you, then choose a different option.
- 5) Once the template is created, click the **Edit Google Doc** button, and you will be taken to the Google Docs page.

** Please Note that Simple Signature does not support merging with documents in other formats, like Microsoft Word. Google Docs nicely supports these documents, but we cannot use them in Simple Signature. In Google Docs, go to File → Save as Google Docs to convert it.*



- 6) You can edit the document as necessary, paste in text, build tables and add graphics or whatever you need. To add in Salesforce data, use the **Show Fields** button. This will open a panel allowing you to browse to specific fields, or related lists. When you select one of these fields, you can copy a “Merge Code” that you can then manually paste into the document. The next section specifies many different merge code options.
- 7) Buttons are also available for The Signature image itself, and a system timestamp that will provide a date/time when the document as merged.
- 8) Once you have created the Google Doc Template, Click on the “Back” button and copy the Record Id of the template (As shown in the image below) and add it to the parameter “Templates” either in the Experience or in the Action button where you have created the Signature Request



Example Merge Codes

The following examples are given to show the versatility of the merge code options. In order to best use these samples, start with pasting the merge code into the document using the Show Fields tab, but then add **options and filters** to them as necessary.

- **Signature Image, No Filter**

```
<B>signature__Signature__c-300</B>
```

- **Signature Image, Filter: When Status = Signed**

```
<B>signature__Signature__c-  
300%<signature__Signature__c><signature__Status__c><Signed></B>
```

- **Normal User Countersignature, No Filter**

```
<B>User.LibrarySignature-300</B>
```

- **5 Days After Today in the USA, No Filter**

```
<B>system.today<day#5><en-US></B>
```

- **5 Days Before Today In India, No Filter**

```
<B>system.today<day-5><hi-IN></B>
```

SIMPLE SIGNATURE FOR SALESFORCE

- The Signature Date In India, No Filter

```
<B>signature__Date__c-<hi-IN></B>
```

- The Signature Date In USA, No Filter

```
<B>signature__Date__c-<en-US></B>
```

- 6 Days After Now In USA, No Filter

```
<B>system.now<day#6><en-US></B>
```

- 6 Days After Now In India, Timestamp, Filter: Firstname = “Testing”

```
<B>system.now<day#6><hi-IN>%<signature__Signature__c><signature__Name__c><Testing></B>
```

- Signature Date in USA, Filter: Firstname = “Testing”

```
<B>signature__Date__c-<en-US>%<signature__Signature__c><signature__Name__c><Testing></B>
```

- Signature Date in USA, Filter: User Is Active = TRUE

```
<B>signature__Date__c-<en-US>%<User><IsActive><true></B>
```

- User Counter Signature, Filter: Status = “Signed”

```
<B>User.LibrarySignature-300%<signature__Signature__c><signature__Status__c><Signed></B>
```

- Custom User Rich Text Field, Filter: User Is Active = TRUE

```
<B>User.User_Signature__c-300%<User><IsActive><true></B>
```

- Signature Image Field with SIZE, Filter: Firstname = “Smith”

```
<B>signature__Signature__c-250%<signature__Signature__c><signature__Name__c><Smith></B>
```

- Signature Image Field Proper Filter User Filter Right Value

```
<B>signature__Signature__c-250%<User><IsActive><true></B>
```

LOCALIZATION & FORMATTING

For Date, Currency, Number and Geolocation fields, tags will automatically be provided when using the field tools. This is the format for these tags, as described in the sorting section:

- `-<LOCALE><CURRENCY>`

When using multiple tags, the localization ones should come first, followed by filter and then sorting tags. The initial dash (“-”) is needed only at the beginning of all tags. For example, the field below uses all tags:

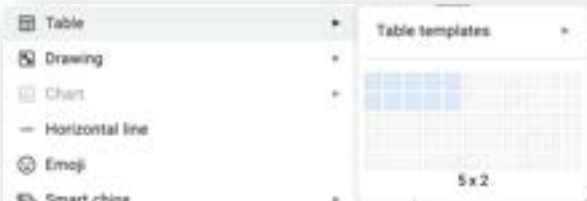
```
<B>signature__Opportunity__r.OpportunityLineItems.TotalPrice<-en-US><USD><#FIL-DEC-3600><#SORT-UP></B>
```

Number fields can also be formatted in a more controlled method. By default number fields will often round up the decimals to provide a cleaner visual, but sometimes you want all the decimals. And in that case, you can specify the format immediately before the closing “” tag, with a #0.0# designation as in this example:

```
<B>signature__ActualSignedLocation__Latitude__s<-en-US><#0.00000000#></B>
```

Related List

Lists of related child-data can be included in Google Docs templates. These lists can be added in one of two ways:



- 1) By selecting the related list when using the same “Show Fields” button. When you do this, you can then select which fields of the child object you would want to display in a list. In the example shown here, you could select the “Opportunities” pointed to by the blue arrow.

However, you do not have any formatting or filtering options for the table, so we do not recommend this option.

- 2) By selecting the “Opportunity >” option, the one with the “>” on it, it would then take you into the Opportunity field list (in this case, Opportunity is the child object). This will give you the merge codes for the child object fields. And these fields can then be used as follows:



SIMPLE SIGNATURE FOR SALESFORCE

- Create a Table 2 rows high and with as many columns as you would like, using the standard Insert → Table menu in Google Docs.
- Now, for each column, give it the column title on the top row, and the field merge code in the second row.

Name	Stage	Owner	Amount
<β>Account__r.Opportunities.Name</β>	<β>Account__r.Opportunities.Stage</β>	<β>Account__r.Opportunities.Owner__c</β>	<β>Account__r.Opportunities.Amount<en-US><USD></β>

In addition, lists in the Google Docs Templates can be filtered and sorted using the codes in this section.

SORTING

A tag should be added to the end of a single Cell in the column that should form the basis for sorting of the whole table. This must be used only one time, it is not currently supported to sort by multiple columns. This is the format for the tag, notice that it must start with “<” and end with “>” and if it is the first tag in the cell, it must be preceded by a “-”:

- <#SORT-UP>
- <#SORT-DOWN>

FILTERS

A tag should be added to the end of a single Cell in the column that contains data to be filtered on. The entire table will be filtered based on the value in this tag. This is the format for the tag, as described in the sorting section:

-<#FIL-TYPE-OPERATOR-VALUE>

- #FIL** - This is the indicator that this is a filter tab
- TYPE** - This specifies the type of field that this column contains. Possible values are:
 - DEC** - for Decimal Numbers
 - INT** - for Integer Numbers
 - BOL** - for Booleans (checkboxes)
 - STR** - for Strings
 - DT** - for Dates
 - DTT** - for Date/Times

SIMPLE SIGNATURE FOR SALESFORCE

- **OPERATOR** – This is the operator to use that will compare the values in this column to whatever your filter is. We cannot unfortunately use standard operators, so use the chart below to see the meaning of each operator along with an example of how it would be set up in the tag.
- **VALUE** – This is the actual value you are comparing to. It is the only optional component. If left empty (the preceding “-” is still needed), then the value will be Blank or Null.

FIELD TYPES	OP	MEANING	EXAMPLE
Boolean, String, Date or Number	EQ	=	<#FIL-STR-EQ-Sample>
Boolean, String, Date or Number	NEQ	!= or <>	<#FIL-STR-NEQ-Sample>
String, Date or Number	GR	>	<#FIL-DEC-GR-100>
String, Date or Number	LS	<	<#FIL-DEC-LS-100>
String, Date or Number	GREQ	>=	<#FIL-DEC-GREQ-100>
String, Date or Number	LSEQ	<=	<#FIL-DEC-LSEQ-100>
String	STWH	Starts With	<#FIL-STR-STWH-Sample>
String	ENWH	Ends With	<#FIL-STR-ENWH-Sample>
String	CTNS	LIKE/Contains	<#FIL-STR-CTNS-Sample>

Two Date Literals, **TODAY** and **YESTERDAY**, can be used instead of actual values when comparing Date or Date/Time columns.

Dynamic Sections

Entire paragraphs of text or other content can be dynamically included or excluded using simple commands, which either hide, or display the content, depending on a Salesforce field. Start and End tags must be placed around the section you want to be dynamic.

Before the dynamic content, place a **CONDITIONSTART** tag followed by a Boolean field that if true, will display the content, but if false, will hide it:

```
<B>CONDITIONSTART<Display__c><true></B>
```

And then, at the end of the dynamic content, place a simple **CONDITIONEND** tag:

```
</B>CONDITIONEND</B>
```

Visualforce PDF Document Creation

If you do not want to use Google Docs, are not using a Document Generation app, and simply prefer using your own document in order to achieve your desired result, use this section to get started building your own document in Salesforce.

The first thing to do is to set up a Visualforce page that will be used as the PDF where the final signature will be merged onto. This Visualforce page will usually be formatted in a particular way, with the logo and other information of your company. This is the most difficult part of the process, and it cannot really be explained in a step-by-step tutorial. If you do not already know how to create Visualforce pages, then you must either ask someone internally to do it, or you can contact us at support@toafinish.com in order to do it for you. We offer hourly services for this type of work.

To get you started, you can use some of the sample Visualforce PDF pages that are included in our GitHub Repository.

GitHub.com Simple Signature Sample Code Repository

<https://github.com/ToAFinish/simplesignature>



Following is a list of the PDF Visualforce page templates currently in the “pages” folder.

File	Description
CustomEmbeddedSignature	This page is to be used instead of the Visualforce page that is embedded in the Signature object page layout.
CustomSignature_hardcoded	Signature page to be used inside of Salesforce for In-Person signatures.
Custom_SiteSignature	Signature page to be used for completing Signature Requests in a Salesforce Site.
SampleAgreementPDF	Simple PDF document laid out like a legal document such as an NDA.
SampleOpportunityPDF *	Slightly complicated PDF which functions as a "quote" from the Opportunity object, including the Opportunity Product lines. *
SampleOpportunityPDF_noController	Version of the Opportunity quote PDF which does not use a controller.
SamplePDF_v2	Copy of the current signature__BasicSignaturePDF that is included with the app. Allows you to update it, change the logo, etc.
SampleQuotePDF *	Slightly complicated PDF which functions as a "quote" from the Quote object, including the Quote Item lines. *

** If you are using Opportunities or Quotes as the parent object, we provide sample Visualforce pages you can use as a starter. Use these instructions to build it in a Sandbox or Dev org. You will likely not be able to do it directly in Production, but rather, you need to do it in a Sandbox and then deploy it to Production.*

- 1) First you will need to add a static resource to your org that allows the signature process. Go to Setup → Custom Code → Static Resources and press the "New" button. Call the Static Resource "sampleinvoice", change the Cache Control to

"Public" and then select a file called **SampleInvoice.zip** which you will need to download from the "resources" folder in our GitHub repository.

- 2) Next, create controller for your Visualforce page. Go to Setup → Custom Code → Apex Classes and press the "New" button. Note that this will only work inside of a Sandbox or Development org. You will not be able to do it directly in Production, so if you do not see a "New" button, that is probably the reason.
- 3) Go back to the GitHub repository, in the "classes" folder, and open the **SampleOpportunityPDFController** or **SampleQuotePDFController** file and copy all of the contents, without the line numbers. Paste all of it into the Apex Class Salesforce window and Save.
 - *You should also copy the matching "Test" file as another Apex Class if you intend to deploy it into production.*
- 4) Finally, you will now need to create a new Visualforce page. Go to Setup → Custom Code → Visualforce Pages and click the "New" button.
- 5) Once you see a new blank 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.
- 6) Go back to the GitHub repository and this time go into the "pages" folder and click on the SampleOpportunityPDF or SampleQuotePDF 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.
- 7) 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.

Once you are finished creating the page, you will need to assign it as the one to use in your Org. To set the default, use the **pdftemplate** parameter in the Signature Experience as discussed in chapter 2. (Remember that the value of the parameter will be the Visualforce page name, the name must be pre-pended with "c__".)

Integration with other Document Generators

If you have a document generation app, like Salesforce CPQ s-Docs, COOLDOX, Conga Composer or Drawloop, a few steps are key to using Simple Signature with your document templates.

- 1) If you are generating a document that can only pull values from a single object, then you need to set up a flow (or signature) that will copy the Signature information *from* the Signature object *into* a field in this object.
 - a. First, we need to create those fields: go to the object you will be using to generate documents from. Go to Setup → Object Manager and click on the object name. Now create the following fields:

Field	API Name	Field Type
Signature Image	SignatureImage__c	Rich Text Area(32768)
Signed	Signed__c	Checkbox

- b. Go to the **Signature** object (Setup → Object Manager → Signature) and create a lookup field to the parent object (the object in step #1.) You can call it what you want. But do not duplicate lookup fields.
- 2) After you have created the fields, then also create a **Flow** that will watch the Signature object for any record that is changed to “Signed” **Status**, and then copy the signature__Signature__c field, and perhaps others like signature__Date__c as well, so that the other objects will have a copy of the signature information in order to merge these fields into the document.
- 3) Next, update the template in your document generator where you want to include a signature, and drop in the Signature image field you created. This field will be blank if it hasn't been signed yet.

Once you have this set up, the following steps need to be performed by someone who is very familiar with the document generation app you are using:

- 4) The first thing that needs to be done is to create a trigger that will take a generated document and create a Signature Request for it. To get you started, you can use a sample code that is included in our GitHub Repository, in the “triggers” folder. We have **CreateSignatureOnAttachments** and **CreateSignatureOnFile**. Both of these assume that the parent object is called Quote__c, so you will need to change that, at least.

GitHub.com Simple Signature Sample Code Repository

<https://github.com/ToAFinish/simplesignature>



- 5) Now, when you generate the document as you typically would from your document generation app, this trigger should create a Signature record as a

Signature Request, so that an email is sent with the document to be signed. If everything is configured correctly, then the signature request will go out via email.

- 6) After the signature request is processed and the Signature record is now Signed, you will need to re-generate the document using whatever tools are provided for automatically generating documents by your selected document generation app. Sometimes this will need to be done via an Apex trigger, other times it might be accomplished through a Flow. Whatever triggers the re-generation, this time the document should have the Signature image in it.

Let us know if you have any questions when going through this section. It can be difficult, especially for the less technical among us. If you need help, reach out to us at support@toafinish.com and we will be happy to help.

That's it. You are ready to test Simple Signature with your document generator.

6 – Administration (paid)

Miscellaneous Stuff for the People Who Keep Everything Going

Configuration of Simple Signature is intentionally easy (simple) and our support team is here to help you with anything that you might run into. This section is intended to answer questions for Salesforce Admins who might need to make large-scale configuration changes, or answer questions for their users.

If you do not find the answers in this implementation guide, you can still reach out to support@toafinish.com to get help, usually within 2 hours during standard business time.

Signature Maintenance

One of the new features in the app is the ability to send reminders on overdue signatures, and perform cleanup operations on a scheduled basis. The new “Maintenance” tab in the Configure page (Setup → Installed Packages → Configure) provides the ability to set up several different types of scheduled tasks:

- **Re-Request** – this option starts the process over for any signatures that meet the criteria. The recipients of the signature request would get an email asking them to sign, just as they would have gotten it when the signature request was just created. Any advancement in the process (for example, if they previously clicked on the button to sign, but did not sign) would be erased, starting over.
- **Notification** – this option simply sends a reminder to the signature request recipient but does not alter the status of the request.
- **Deletion** – this option will delete signature requests that meet the given criteria. Can be used for old signature never signed, or for requests that are already signed and no longer needed.
- **Clean Up** – this option will delete related records no longer needed for records that meet the given criteria. Process records, preview File PDFs, and sensitive values, such as a password would all be cleared out.

The screenshot shows a 'New Maintenance Rule' form. The fields are filled with the following values:

- Name:** Reticent NDA Signers
- Type:** Notification
- Status:** Viewed
- Days:** 10
- Request Form:** NDA
- Email Template:** Please_Sign_NDA_Already

Buttons: Save, Cancel

- 1) To set these up, go to the “Maintenance” tab, and click the **New Maintenance Rule** button. Enter the following values:
 - a. Give it a descriptive **Name** indicating what the Maintenance Rule will do.
 - b. Select a **Type** using the descriptions provided above.
 - c. Select a **Status**. This will restrict the current rule to only work on signature requests with this status. And yes, if you want a rule to apply to multiple statuses, you will need to create multiple rules. “Deletion” and “Clean Up” Types will typically use the “Signed” status, while other Types would use earlier statuses.
 - d. Enter a number of **Days** that must have gone by from when the Signature Request *entered* the status you selected, before this maintenance rule would take effect. For example, if you select “Emailed” as the Status, and “3” as the Days, then the maintenance rule would take effect on the 4th day (after 3 days) when a Signature Request was Emailed and has not yet been Viewed or Clicked or Signed (it is still in “Emailed” status after 3 days).
 - e. Enter a value in the **Request Form** field in order to further filter which Signature Requests are processed in the maintenance rule. This filter is intended for installations where multiple signature request processes exist, and you want to run the current rule on only one of those processes. The RequestedForm__c field on the Signature object will be used to compare the value in this field vs what is entered in this filter. (This field is typically set using the “requestedform” parameter in the Signature Experience.)

- f. Two Types of Maintenance Rules involve emailing (Re-Request and Notification) and for those, if the **Email Template** field is populated with the name of an Email Template, then this email template would be used. If not, the default email template is used (the same one that would have been used when originally sending out the Signature Request).
 - g. For Maintenance Rules which involve emailing, the **CC Record Owner** option will copy the record owner of the Signature, typically the user who created the Signature Request.
 - h. For Maintenance Rules which involve emailing, the **CC Record Owner's Role Hierarchy** option will copy the Role-based hierarchy above the record owner of the Signature (typically the user who created the Signature Request) on the email sent. This might go several layers up in the reporting structure and should be used with care.
 - i. For Maintenance Rules which involve emailing, the **BCC Record Owner's Email to Salesforce** option will copy the "My Email to Salesforce" email address for the record owner of the Signature, typically the user who created the Signature Request. This email address attempts to create a history record for the email sent, attached to the current record and record owner. (This must be set up in advance. Go to My Settings → Email → My Email to Salesforce.)
 - j. Click the **Save** button are you are done creating the rule.
- 2) Now, you must schedule the Maintenance Job to run. When it runs, it will run on all maintenance rules, so make sure you have reviewed anything you have set up, to ensure it will do what you want it to do. Once you have reviewed, go to Setup → Custom Code → Apex Classes and press the "Schedule Apex" button and for the Apex Class select **SignatureCleanupScheduler**.

** Please be careful when scheduling this class, it will potentially do things on a fairly massive scale – email a bunch of clients or delete records that might not be recoverable. Make sure you know what you are doing before you schedule this class to run. We would recommend having our support team take a look at it for you the first time you set it up.*

- 3) That should do it!

Signature Report/Bulk Printing

☐ In Lightning, click on the **Search Apps and Items** icon on the top-left of the page, and type in Signature Report in the search box. This report allows you to run a PDF of each signature on a different page. It includes filters to specify the dates and types of Signatures that should be included in the PDF.

** Please note that this Report PDF is somewhat basic and intended only for backup of signatures. It is not intended for distribution or client-facing applications.*

Multiple Salesforce Orgs

As a general rule, Simple Signature must be configured in each org where you want to use it. Once you have fully configured a Production Org, any Sandboxes you refresh or create from Production will be mostly configured, with a few exceptions, defined below.

Sandbox Configuration

If you install Simple Signature into a Sandbox, it must be configured from the beginning, just like a Production Org. However, if you are working in a new sandbox that has been refreshed or created from Production, there are only a limited number of things that must be done, especially if you are using Signature Requests:

- 1) Switch to Classic. This might break in Lightning, depending on your release.
- 2) First, the Site URL for the new sandbox needs to be set. Go to Setup → User Interface → Sites and Domains → Sites and click on the “Site URL” of the Site that is used for Signatures. Once this site opens in a new tab, copy the URL to the clipboard.
- 3) Now, go to Setup → Installed Packages → Simple Signature → Configure and click on the **Configuration** tab, and paste the URL from step #1 into the “Site URL for Signature Requests” box (make sure to include a slash “/” at the end).
- 4) Scroll down to the “Processing Options” and make sure the **Turn Off Emails** option is unchecked IF you want to send emails in this sandbox. Please note that having this option unchecked while doing large-scale testing, or while running Maintenance processes, could
- 5) mode to do this.) It will prompt you to accept the rights for the Site access, result in emails being sent out from this sandbox to real email addresses from your customers. If this is a concern, especially if you are in a Full, or Partial sandbox, please leave this option Checked, and then use the **Blacklist and Whitelist Emails/Domains** to whitelist your own test email addresses, enabling you to receive emails, but blocking other emails from going out.
- 6) Once configured, press the green **Save Settings** button.
- 7) Next, still in the Configure page, click on the **Site Configuration** tab. You will likely see a notice about remote settings, and also a “Re-Authorize” button. (You might have to be in Classic and also likely require you to re-log into the sandbox again. Once you do this, you will see the same tab again, with a different access URL, similar to the screenshot below:



- 8) Copy the URL from the “To Enable Site Access” message, and then click on the link to go to the **Remote Site Settings** list. Once there, add a new Remote Site with the URL provided. The “Remote Site Name” can be anything you like, but the “Remote Site URL” should be exactly the URL provided in the previous screen. Make it “Active” and **Save**.
- 9) If you are using Google Docs, you will also need to re-authorize the Google Docs integration. Go back to the Configure page (Setup → Installed Packages → Simple Signature → Configure) and click the Revoke button on the **Google Doc Configuration** tab. When you do, it should open up a popup window, asking you to log into your Google account again:
 - a. Select the Google Account you want to log into for Google Docs.
 - b. Grant the rights it asks for on the next screen.
 - c. Copy the Authorization Code from the final popup screen and then paste it into the “Authorization Code” field and press the Submit button.
- 10) If you are NOT in a Partial or Full Sandbox, and are using Google Docs, you will also need to create a Template record for each of the documents that you want to test, unfortunately. It is fairly easy to do so, but you need to go to the “ β Templates” object in both Production (or wherever your templates are already set up) and the Sandbox. You can view “All” in Production, and for each one, you need to do these steps:
 - a. Click on the Template in production.
 - b. Copy the “Google Doc ID” value.



- c. Go to the Sandbox and click “New” to create a new Template. Select “Google Docs” and press **Next**. Then choose “Create New Template” and press **Next** again.
- d. On the New Template page, paste in the Google Doc ID and then enter all the other fields to match what is in Production.
- e. **Do NOT** check the “Copy Template” option. Save.

- f. When the record is created, copy the **Salesforce ID** for the template. You can find this in the browser's URL. It will be an alphanumeric value either 15 or 18 characters long.
 - g. Now you need to paste this Template's Salesforce ID in any place where you are calling this Signature Request from. Typically, this would either be an Action, or in a Salesforce Experience.
- 11) Finally, you will likely need to turn on outgoing email. Go to Setup → Email → Deliverability, and set the Access Level to "All Email" and **Save**.

Note that doing Step 9 might be dangerous if your sandbox contains data (if it is a Full or Partial sandbox) because automated processes or user mistakes could send emails to unsuspecting Contacts or Leads in your sandbox. If this is a likely risk, then we recommend only turning on this option when you are testing, and only if you know that no automated processes are going to run and send emails to customers.

Production Deployments

If you install Simple Signature into a Sandbox, and after testing, want to deploy to your Production Org, there are a few things to consider.

- 1) You cannot deploy most of the basic settings to Production. These items must be manually re-configured in Production to match what you have in the Sandbox:
 - a. App Installation of the Simple Signature app must be done manually in Production. You can install it from the AppExchange listing (search for "Simple Signature") or, better yet, you can ask the support team for a direct install link.
 - b. App version. Often you will be on a different app version in a Sandbox than you are in Production. You can see the version at the bottom-right of the Configure page, or by going to Setup → Installed Packages and looking for the Simple Signature package listing. If the versions are different, ask the support team to push out the new version to Production, or ask for a link to do it yourself.
 - c. If using Signature Requests, the Salesforce Site must also be set up in Production as described in the next chapter. The "Site Configuration" steps must be completed in Production.
 - d. If using Google Docs integration, the Google Configuration in chapter 5 must also be completed in Production.
 - e. Any settings on the Configuration tab of the Configure page (Setup → Installed Packages → Simple Signature → Configure) will need to be re-entered and saved. The only one that *must* be different is the **Site URL for Signature Requests**, if using Signature Requests.

- f. On the Experiences tab in the same Configure page, you can actually use the “Export” button in a Sandbox, and then the “Import” button in Production, to bring over the Experiences. However, please note a few things:
 - i. The Experience IDs will be different in Production than they were in the source Sandbox. This means that if you hard-coded an Experience ID anywhere, you will need to modify it.
 - ii. Some of the parameters in an Experience use commas, and the **Import does not support this feature**. For example: fieldreadonly, hideblankfields, templates, and submissiontype parameters, can all have a list of field API names, separated by commas. This means that after an import, careful review should be made in order to modify any of these parameters to the correct specifications.
 - iii. When using the Import feature, you want to select the Fields and Field Type as shown in the picture below.

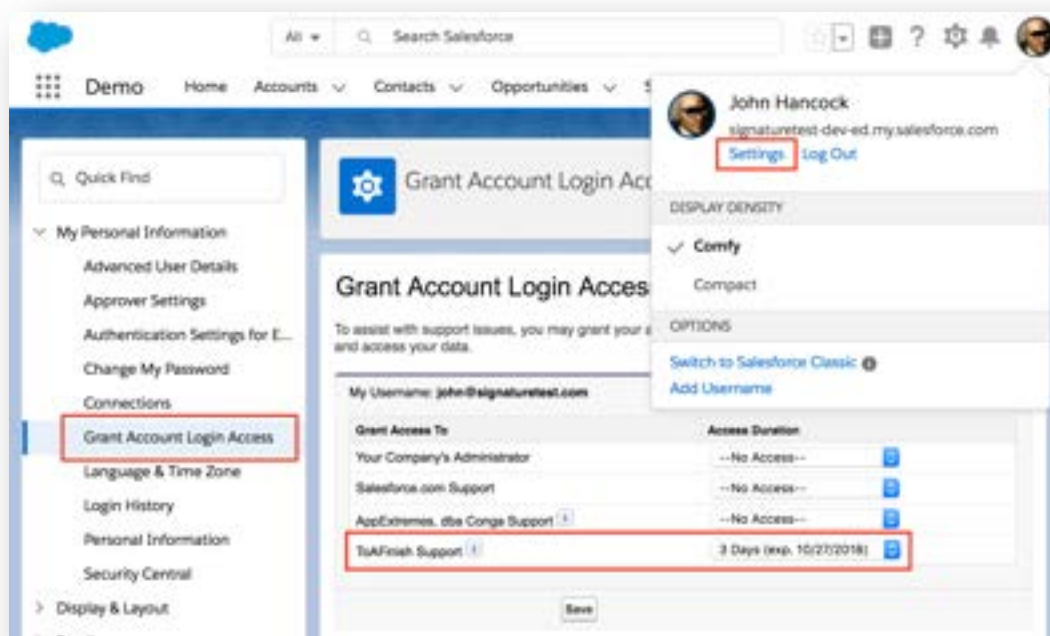
Column Name	Field	Field Type
Experience Name	Experience Name	Experience
Description	Description	Experience
No. of Params	Number of Parameters	Parameter
Param Name	Parameter Name	Parameter
Field Value	Field Value	Parameter
Param	Param	Parameter
Value	Value	Parameter

- g. Finally, you will need to either re-create or deploy via a change set, any custom fields, pages, Lightning components, actions, or other customizations that you have created in the Sandbox. To A Finish technical support can help with this deployment if necessary.

Good luck!

Feedback and Support Access

Again, 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 support 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 an 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=a0N3A00000DqCoUUV>

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.