



C R E D E R A

BrightSIGN Implementation Guide

Version 4.0



- 1 Overview
- 2 Configuration
- 3 Usage
 - 3.1 Aura Lightning Component/Application
 - 3.2 Lightning App Builder
 - 3.3 Community Builder
 - 3.4 Lightning Flow
- 4 Extending
- 5 Signature Captured Event
- 6 Multiple BrightSIGN Components on a Single Page
- 7 Guest User Access
- 8 Troubleshooting



1 Overview

Signature Capture is now BrightSIGN!

BrightSIGN is a Lightning Component that allows a user to draw their signature into an HTML5 canvas element via a browser or mobile device, convert the captured drawing to an image and store that image against a Salesforce record.

This guide is for version 4.1+ of the SignatureCapture managed package.

2 Configuration

BrightSIGN is a public Aura Lightning Component that may be used programmatically or as part of a low/no-code Lightning Flow. It implements the interfaces required to be used in drag and drop tools such as the Lightning App Builder and is tested on the Salesforce Lightning desktop and mobile applications.

The behaviour of BrightSIGN is configurable via attributes which are available directly for programmatic applications, and as design attributes for low/no-code tools.

The component supports the following attributes:

Parameter	Description	Default
<code>startMsg</code>	Message to display to the user to start the capture process	Click the 'Capture Signature' button to begin
<code>enterMsg</code>	Text to display above the BrightSIGN canvas	Sign here please
<code>completeMsg</code>	Text to display above the image of the captured signature once it has been saved to the server	Here is the captured signature
<code>recordId</code>	The id of the Salesforce record to attach the signature image to	None - this will be supplied by the Salesforce platform when BrightSIGN is embedded in a record page.
<code>saveAttachment</code>	Should the image be saved as an attachment (true) or a file (false)	true
<code>width</code>	Width of the BrightSIGN canvas	100%
<code>height</code>	Height of the BrightSIGN canvas	300px
<code>captureButtonLabel</code>	Label for the capture signature button	Capture Signature
<code>saveButtonLabel</code>	Label for the save button	Save

<code>clearButtonLabel</code>	Label for the clear button	Clear
<code>skipToCapture</code>	Skip straight to capturing a signature without prompting the user or waiting for a button click	false
<code>fileName</code>	File name for BrightSIGN	Signature.png
<code>backgroundColour</code>	Background colour for image	White
<code>orphanImageFile</code>	Save the signature image as an orphaned File – this allows a Guest user to save a signature	false

3 Usage

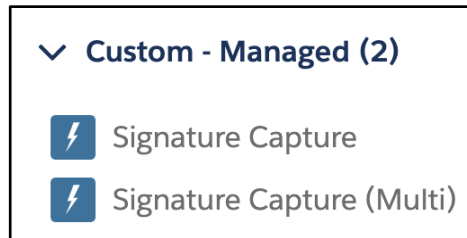
3.1 Aura Lightning Component/Application

BrightSIGN can be embedded programmatically inside an Aura component or Application as follows:

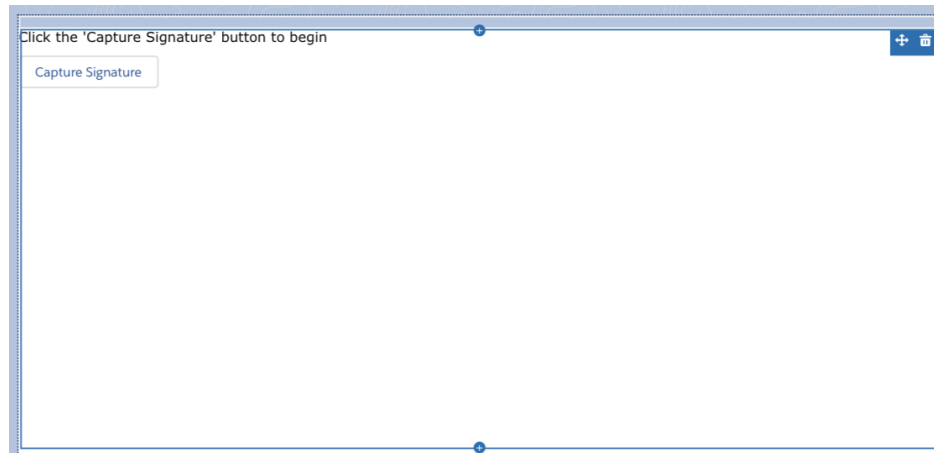
```
<BGSIGCAP:SignatureCapture
    recordId="{!v.recordId}"
    startMsg="Click the button to get started"
    enterMsg="Please sign below"
    completeMsg="Here is the signature you entered"
    width="50%"
    height="200px"/>
```

3.2 Lightning App Builder

BrightSIGN appears in the list of available components in the app builder, in the Custom - Managed section at the bottom of the Components list of the left hand side of the page.



Dragging the component onto the canvas displays the component with the default attribute values.



The values may be changed by clicking on the component, which opens the attribute editor on the right hand side of the page.



Page > Signature Capture

Start Message i

Click the 'Capture Signature' button to begin

Enter Message i

Sign here please

Complete Message i

Here is the captured signature

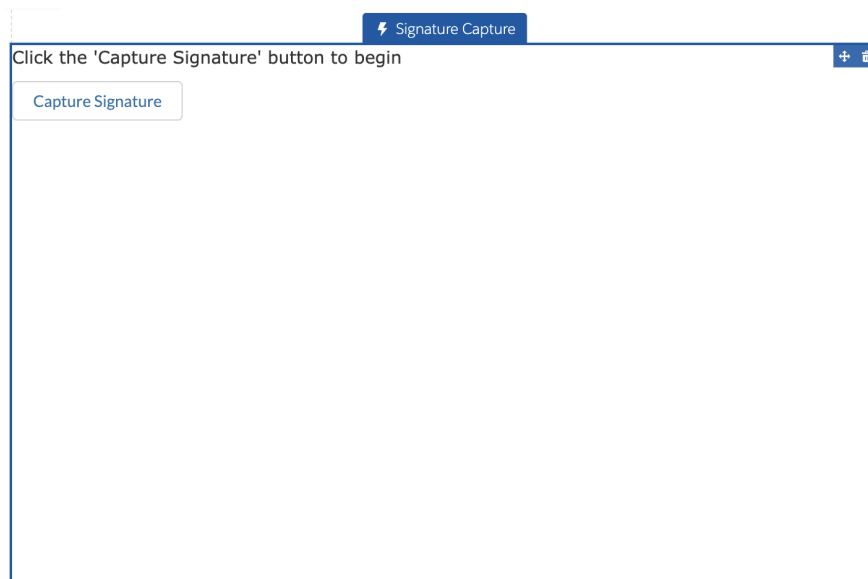
If the component is added to a Record page, the Salesforce Platform will automatically populate the `recordId` attribute with the ID of the record being viewed. For Home or Application pages, the record id will need to be manually entered.

3.3 Community Builder

BrightSIGN appears in the list of available components in the app builder, in the Custom - Managed section at the bottom of the Components list of the left hand side of the page.



Dragging the component onto the builder canvas displays the component with the default attribute values:



Clicking on the component opens the attribute editor on the right hand side of the page:



Signature Capture ▾ ✕

Start Message ⓘ

Click the 'Capture Signature' button to

Enter Message ⓘ

Sign here please

Complete Message ⓘ

Here is the captured signature

* Canvas Width ⓘ

100%

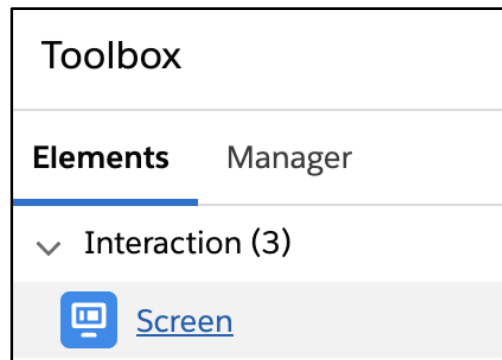
* Canvas Height ⓘ

300px

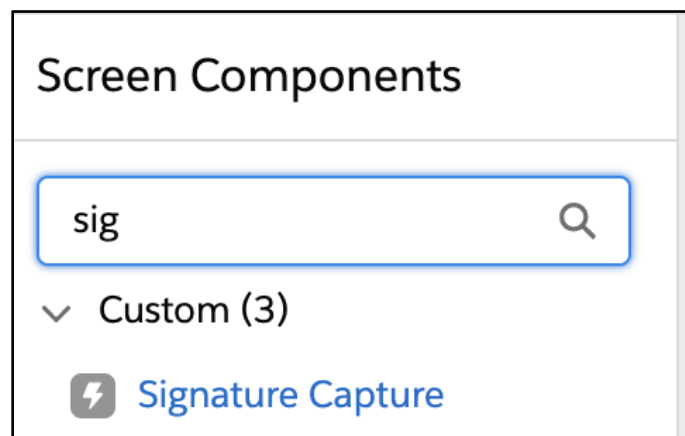
If the component is added as a direct descendant of a Record page, the Salesforce Platform will automatically populate the `recordId` attribute with the ID of the record being viewed. If nested inside another component in a Record page (a tab, for example) then you will need to specify the `recordId` attribute as `: {!recordId}`

3.4 Lightning Flow

To use BrightSIGN in a Lightning Flow, drag a screen element from the toolbox to the canvas:



In the resulting dialog, type 'sig' into the search box on the left hand side:



Choose BrightSIGN from the list of results and this adds the BrightSIGN to the screen with the default attributes.



Specify the API name and change any other attributes as desired.

The screenshot shows the configuration interface for the 'Signature Capture' component. On the left, a flow editor shows a component box with a lightning bolt icon and the text 'Signature Capture' and 'No preview is available for this component.' Below the component are 'Pause', 'Previous', and 'Finish' buttons. On the right, a configuration panel titled 'Signature Capture' lists the following settings:

- * API Name:
- * Canvas Height:
- * Canvas Width:
- Background Colour:
- Capture Button Label:
- Clear Button Label:

Launching the flow shows the BrightSIGN component:

The screenshot shows the 'Signature Capture' component in a flow. The component has a title bar 'Signature Capture' and a main area with the text 'Sign here please' above a large empty rectangular canvas. At the bottom of the canvas are 'Save' and 'Clear' buttons. At the bottom right of the component is a 'Finish' button.

Note that by default the screen component displays the navigation footer - see the samples repo for a solution that handles the navigation automatically.



4 Extending

The functionality provided by BrightSIGN can be extended by wrapping it inside other Aura components. As the extended functionality tends to be specific to a customer scenario, these wrapper components are not added to the BrightSIGN package on the app exchange. Instead they are made available via a samples repository on Github at :

<https://github.com/keirbowden/sigcapapp>

These components are suitable for deployment to a subscriber org (the Salesforce instance that you have installed BrightSIGN) and are open source so may be copied and extended under the permissive terms of the [MIT License](#).

Many samples are accompanied by a blog post that explains how it works in detail.

5 Signature Captured Event

When a signature is successfully captured against a record, the `SignatureCapturedEvt` Aura application event is fired. This contains the Id of the record in the `recordId` parameter. The `SigCapNotifier` component in the samples repository demonstrates how to consume this event to display a toast message.

6 Multiple BrightSIGN Components on a Single Page

New in version 2.0

Version 2.0 of BrightSIGN introduces the ability to embed multiple components to capture a signature on a single page. As this would have required significant changes to the existing component architecture, a new component has been produced to handle this scenario named `SignatureCaptureMulti`. This can be used in all of the scenarios shown in the [Usage](#) section and supports the same attributes as the `SignatureCapture` component. It also supports an additional attribute of `name` - a text string that uniquely identifies the component on the page. It is recommended that the name of the captured Attachment/File (the `fileName` attribute) is changed from the default to a unique value for each instance of `SignatureCaptureMulti` added to the page.

When `SignatureCaptureMulti` fires the `SignatureCapturedEvt` event, it includes a `name` parameter that identifies the instance that successfully captured the signature against the record.

7 Guest User Access

New in version 4

Version 4 of BrightSIGN adds support for unauthenticated Guest users to save signature images, with the following caveats:

- The image will not be linked to a record
The user must have Update access to the parent record in order to link the image, but this is not supported for Guest users. Instead, the image will be saved as an orphan and only linked to the user who created it
- The image must be saved as a File
Orphaned attachments are not supported

To allow a Guest user to save a signature image, set the `orphanImageFile` parameter to `true`.

8 Troubleshooting

Problem: When saving the signature, users receive the error message :

'You do not have update permission on the parent subject type for record with id .'

Solution: Assign the user explicit edit permission to the parent subject type you are adding the Attachment/File to via their Profile or a Permission Set.

Explanation: In order to pass the security review, any Apex code must demonstrate that it is respecting the security configuration of the Salesforce instance. This includes explicit CRUD and FLS checks before carrying out any DML. In order to save a signature image as an attachment or file associated with a parent record, the user must have **explicit** edit permission for the subject type of the parent record via their profile or an assigned permission set.



C R E D E R A



+44 (0) 333 043 9000



www.credera.com/contact