

GimbalBarcode

Barcodes made easy in Salesforce

Configuration Guide

Version 8.14

Table of Contents

About Gimbal Logic.....3

About Gimbal Barcode.....3

Generating Barcode Images4

 Generating a PNG Image of a Barcode (Content File or Base64 String)4

 Example: Create a Barcode by Manually Creating a Barcode Request Record6

 Example: Create a Barcode using Flow or Omniscript.....8

 Example: Create a Barcode by Inserting Records using Apex9

 Rendering a Barcode on a VisualForce Page10

Barcode Scanning using Flow 11

Implementation Assistance and Support 16

About Gimbal Logic

Gimbal Logic is a Salesforce.com ISV Partner. We enable cost-effective native Salesforce solutions for inventory management, attendance tracking, manufacturing, field service, and process automation.

If you would like some help designing or implementing a barcode solution for your Salesforce environment, please contact us at barcode@gimballogic.com.

About Gimbal Barcode

Gimbal Barcode is a package of functionality that can be installed in a Salesforce org, and then used by a Salesforce Administrator to create robust barcoding applications for all users. Gimbal Barcode has two main areas of functionality: barcode **generation**, and barcode **scanning**.

Generating Barcode Images

Gimbal Barcode provides two options for generating barcodes:

- 1) Generate a PNG image that is stored as a Salesforce Content File (i.e. ContentDocument) or a Base64 string.
- 2) Render a barcode within a VisualForce page using a VisualForce Component.

Generating a PNG Image of a Barcode (Content File or Base64 String)

To generate a PNG image of a barcode, simply insert a Barcode Request record (i.e. Salesforce sObject of API name GMBLBC2__Barcode_Request__c). The Gimbal Barcode application will process the inserted record, and either attach the resulting image as a ContentDocument linked to one or more records, or write the Base64 string data to a field on the Barcode Request record, depending on the “Output Type” specified on the Barcode Request record.

Barcode Request records can be created using any of the standard Salesforce mechanisms for creating records:

- Manually, by navigating to the Barcode Request tab and clicking New
- Declaratively, using automation tools such as Flow or Omniscript
- Programmatically, using Apex

Barcode generation is asynchronous. A single barcode request typically completes in under one second, but large batches (e.g. hundreds of barcodes at once) may take longer. Do not assume the OUTPUT fields are populated immediately after the record is saved or inserted — refresh the page and/or check the Status field to confirm processing has completed.

Fields

The table below lists the fields on the Barcode Request object. INPUT fields are populated by the user (or by automation/Apex) to define the barcode to be generated. OUTPUT fields are read-only and populated by Gimbal Barcode once processing completes.

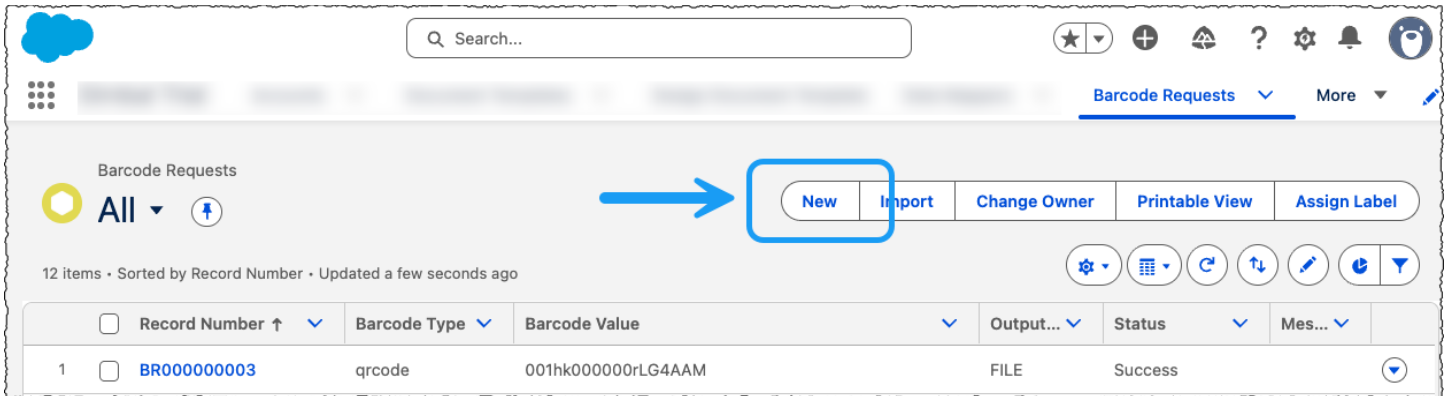
Field	Type	INPUT / OUTPUT	Description / How to Populate
Barcode Type	Picklist	INPUT	Required. The barcode symbology to generate: code128, code39, datamatrix, ean13, gs1-128, gs1datamatrix, interleaved2of5, pdf417, qrcode, upca, or other. Gimbal Barcode supports 100+ symbologies; select other for anything not listed, and consult the documentation or contact support@gimballogic.com.
Barcode Value	Text	INPUT	Required. The data to encode in the barcode (e.g. a SKU, serial number, or URL).
Output Type	Picklist	INPUT	Required. FILE saves the barcode image as a Salesforce Content File. BASE64 returns the image as a Base64-encoded string on the record.
Extra Params	Text	INPUT	Optional. Well-formed JSON specifying additional formatting or content options, e.g. {"includetext":true,"padding":"5","backgroundcolor":"FFFFFF"}.
File Name	Text	INPUT	Optional; used only when Output Type = FILE. Name of the File that will be created. Defaults to "barcode" if left blank.
Linked Record IDs	Text	INPUT	Optional; used only when Output Type = FILE. Comma-separated list of record IDs to relate the File to (e.g. an Account ID, or a public library ID for Salesforce Document Generation).
Status	Picklist	OUTPUT	Defaults to New when the record is created. Gimbal Barcode updates this to Success or Failure once processing completes.
Message	Text	OUTPUT	Populated with an error message if barcode generation fails. Blank on success.

Base64 Image Data	Text	OUTPUT	Populated with the Base64-encoded barcode image when Output Type = BASE64.
Content Document ID	Text	OUTPUT	Populated with the ID of the Content Document created when Output Type = FILE.
Content Version ID	Text	OUTPUT	Populated with the ID of the Content Version created when Output Type = FILE.

Most document generation solutions (e.g. Salesforce Document Generation, Conga, etc.) will require the Content File to belong to a File Library to which the document generation solution has access. Make sure to include the ID of the File Library in the “Linked Record IDs” field.

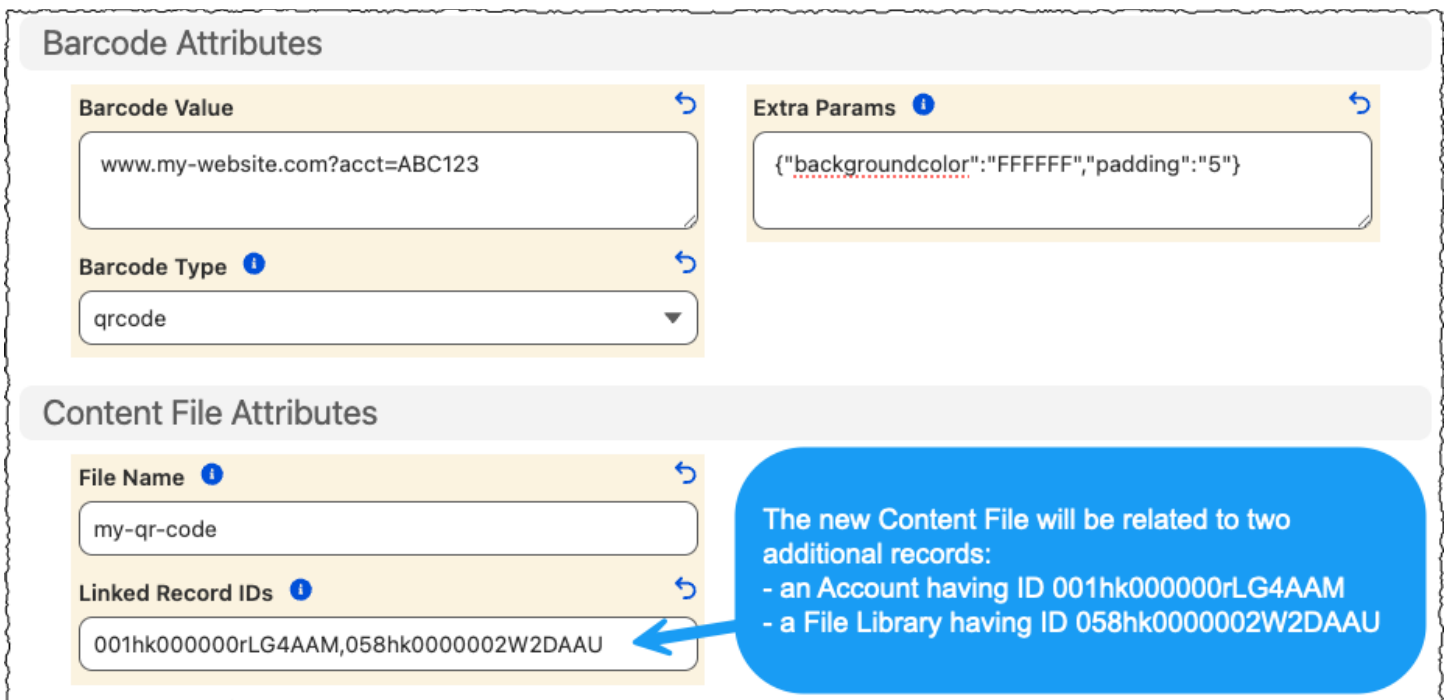
Example: Create a Barcode by Manually Creating a Barcode Request Record

- Navigate to the tab for the Barcode Request object. (If the tab is not visible, add it to an App using the App Launcher, or ask your Salesforce Administrator to add it.)
- Click New.



The screenshot shows the Salesforce interface for the 'Barcode Requests' object. At the top, there is a search bar and a navigation bar with the 'Barcode Requests' tab selected. Below the navigation bar, there is a section for 'Barcode Requests' with a filter set to 'All'. A blue arrow points to the 'New' button in the top action bar. Below this, there is a table with 12 items, sorted by Record Number. The table has columns for Record Number, Barcode Type, Barcode Value, Output..., Status, and Mes... (Message). The first row shows a record with Record Number 1, Barcode Type 'qrcode', Barcode Value '001hk000000rLG4AAM', Output 'FILE', and Status 'Success'.

- Populate the INPUT fields described above (Barcode Type, Barcode Value, and Output Type are required; Extra Params, File Name, and Linked Record IDs are used as needed).



The screenshot shows the 'Barcode Attributes' and 'Content File Attributes' sections. The 'Barcode Attributes' section has two fields: 'Barcode Value' (text input) and 'Barcode Type' (dropdown menu). The 'Content File Attributes' section has three fields: 'File Name' (text input), 'Linked Record IDs' (text input), and 'Extra Params' (text input). A blue callout box points to the 'Linked Record IDs' field and contains the following text:

The new Content File will be related to two additional records:

- an Account having ID 001hk000000rLG4AAM
- a File Library having ID 058hk0000002W2DAAU

- Click Save.
- Barcode generation happens asynchronously. Refresh the record after a moment to see the Status field change to Success (or Failure, with a Message explaining why). The new Content Document will appear in the Files related list, and the ID of the new Content Document and the Content Version will be populated. (If an Output Type of BASE64 was selected, the Base 64 field will become populated instead.) See the screenshot below.



Barcode Request

BR000000014

[New Contact](#)

[Edit](#)

[New Opportunity](#)



Record Number

BR000000014

Status

Success

Status has updated to "Success".

Output Type ⓘ

FILE



Barcode Attributes

Barcode Value

www.my-website.com?acct=ABC123



Extra Params ⓘ

{"backgroundcolor":"FFFFFF","padding":"5"}



Barcode Type ⓘ

qrcode



Content File Attributes

File Name ⓘ

my-qr-code



Content Document ID ⓘ

069hk000000LB6wAAG

IDs of the new Content Document and Content Version

Linked Record IDs ⓘ

001hk000000rLG4AAM,058hk0000002W2DAAU



Content Version ID ⓘ

068hk000000LxwMAAS

> Output

> System Information



Files (1)

[Add Files](#)



my-qr-code

Jul 20, 2026 • 1KB • png

The new barcode image appears in the Files related list of the Barcode Request, and also in the Files related list on the Linked Records.

[View All](#)

Example: Create a Barcode using Flow or Omniscript

To create a Barcode using Flow, use the standard “Create Records” Flow Component to insert a Barcode Request record, as follows:

The screenshot displays the Salesforce Flow Builder interface for a flow named "Generate Barcode 9 - V1". The flow is currently inactive. The flow canvas on the left shows a sequence of three steps: "Screen Flow Start", "Create Barcode Request" (which is highlighted with a blue border and labeled "Create Records"), and "End".

The right-hand pane shows the configuration for the "Create Records" component. The "Label" is "Create Barcode Request" and the "API Name" is "Create_Barcode_Request". The "Description" field is empty. The "How to set record field values" is set to "Manually".

Under the heading "Create a Record of This Object", the "Object" is set to "Barcode Request".

The "Set Field Values for the Barcode Request" section contains five rows, each with a "Field" and a "Value":

- Field: Barcode Type; Value: A_a qrcode
- Field: Barcode Value; Value: A_a recordId
- Field: Extra Params; Value: {"backgroundcolor":"FFFFFF","padding":"5"}
- Field: File Name; Value: Barcode1b
- Field: Linked Record IDs; Value: A_a LinkedRecordIds

At the bottom of this section is a "+ Add Field" button and a checkbox labeled "Manually assign variables (advanced)".

Because barcode generation is asynchronous, a Barcode Request record created earlier in a Flow may not yet have its OUTPUT fields populated (and the new Content File may not have been created) by the next element in the same transaction. If your Flow needs the resulting File or Base64 image data, wait for the Status field of the Barcode Request record to change to “Success” before continuing.

To create a barcode using Omniscript, use a Data Mapper Post Action to insert a Barcode Request record.

Example: Create a Barcode by Inserting Records using Apex

Barcode Request records can be created programmatically with standard DML, just like any other sObject:

```
GMBLBC2__Barcode_Request__c req = new GMBLBC2__Barcode_Request__c(  
    GMBLBC2__Barcode_Type__c = 'qrcode',  
    GMBLBC2__Barcode_Value__c = myAsset.SerialNumber__c,  
    GMBLBC2__Output_Type__c = 'FILE',  
    GMBLBC2__File_Name__c = 'AssetLabel',  
    GMBLBC2__Linked_Record_IDs__c = myAsset.Id  
);  
insert req;
```

As with the other methods, set only the INPUT fields. If necessary, query the inserted Barcode Record(s) after processing completes to retrieve the OUTPUT field values (e.g. Content Document ID).

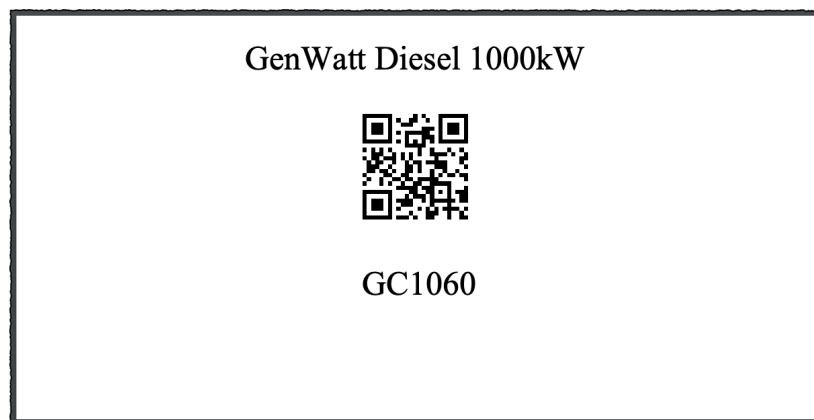
Rendering a Barcode on a VisualForce Page

In order to render a barcode on a VisualForce page, it is **not** necessary to first create a barcode image. The barcode image can be rendered directly in the VisualForce page using the GMBLBC2:barcode VisualForce Component. This is particularly useful for generating labels for physical items, and it can also be used for workflow documents such as packing slips, sales orders, and correspondence such as event invitations.

Here's an example of using Gimbal Barcode to generate a Product label:

```
1 <apex:page applyHtmlTag="false" renderAs="pdf" showHeader="false" standardController="Product2">
2   <head>
3     <style type="text/css">
4       @page {
5         size: {!NULLVALUE($CurrentPage.parameters.pageSize, '4in 2in')};
6         margin:0;
7       }
8     </style>
9   </head>
10  <body style="margin:0;">
11    <div style="text-align:center; margin:3mm;">
12      {!Product2.Name}
13      <br/><br/>
14      <GMBLBC2:barcode value="{Product2.ProductCode}" type="qrcode"/>
15      <br/><br/>
16      {!Product2.ProductCode}
17    </div>
18  </body>
19 </apex:page>
```

When launched from a button on the detail page of a Product record, the following 4" x 2" label will open for printing:



The implementation team at Gimbal Barcode has experience with a wide variety of requirements for barcode label printing using VisualForce. Please don't hesitate to ask for our expert assistance. Contact us at barcode@gimballogic.com.

Barcode Scanning using Flow

Gimbal Barcode includes a Flow component that enables barcode scanning within Salesforce running on a desktop or laptop computer, or a mobile device using the Salesforce Mobile App. Common use cases for barcode scanning within Salesforce include:

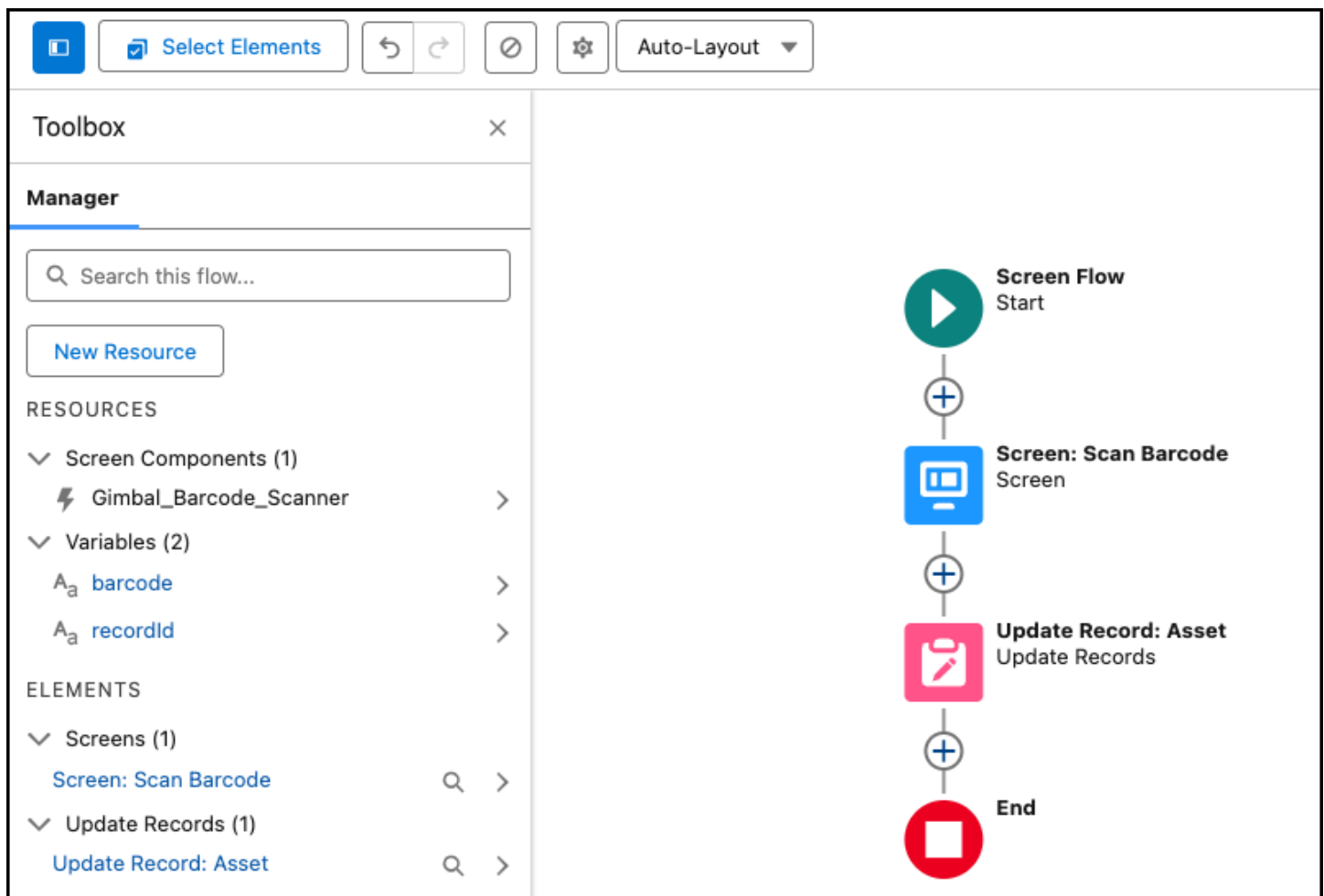
- Inventory Control
- Asset Management
- Attendance Tracking
- Event Management
- Process Control (manufacturing, diagnostic labs, etc.)
- Order Fulfillment

Not all barcode scanning use cases are appropriate for Flow. Some barcode scanning use cases require the creation of a custom Lightning Component. Our implementation team can help you create a custom Lightning Component designed for your specific use case. Contact us at barcode@gimballogic.com.

In order to add barcode scanning to any screen flow, simply drag the component named “Gimbal Barcode Scanner for Flow” onto the canvas.

Here are the detailed steps for creating a sample screen flow whose purpose is to update the Serial Number field of an Asset.

The overall structure of the flow is as follows:



Step 1: Create Flow Variables

Create **two** flow variables, as depicted below.

Variable Name:	recordId
Data Type:	Text
Availability Outside the Flow:	Available for Input

Edit Variable

* API Name ⓘ

recordId

Description

* Data Type ⓘ

Text

☐ Allow multiple values (collection) ⓘ

Default Value

Enter value or search resources...

🔍

Availability Outside the Flow

☒ Available for input

☐ Available for output

Cancel

Done

Variable Name:	barcode
Data Type:	Text
Availability Outside the Flow:	none

Edit Variable

* API Name ⓘ

barcode

Description

* Data Type ⓘ

Text

☐ Allow multiple values (collection) ⓘ

Default Value

Enter value or search resources...

🔍

Availability Outside the Flow

☐ Available for input

☐ Available for output

Cancel

Done

Step 2: Create a Screen and add the “Scanner for Flow” Component

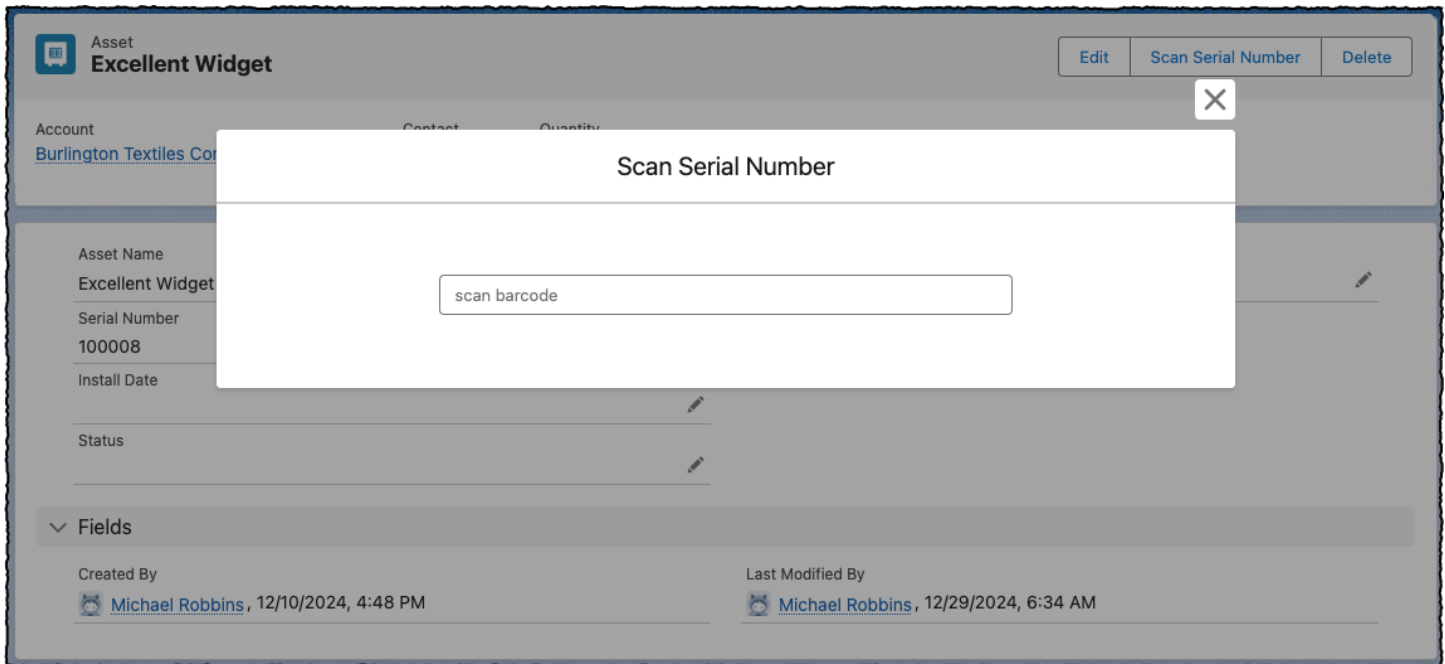
- Drag the component onto the canvas from the Components menu in the left sidebar.
- Specify the barcode variable in the Advanced sub-menu.

The screenshot shows the 'Edit Screen' interface. On the left, the 'Components' sidebar lists various components. The 'Gimbal Barcode Scanner for Flow' component is highlighted in the 'Custom (1)' section. A red arrow points from this component to its instance on the canvas. The canvas shows a single instance of the 'Gimbal Barcode Scanner for Flow' component, labeled 'Gimbal_Barcode_Scanner'. The right sidebar shows the configuration for this component. The 'API Name' is 'Gimbal_Barcode_Scanner'. The 'Auto-Navigate Next', 'Desktop Auto-Focus', and 'Mobile Auto-Launch' options are all set to 'Active'. The 'Set Component Visibility' and 'Validate Input' sections are collapsed. The 'Advanced' section is expanded, showing 'Manually assign variables' checked. Under 'Store Output Values to Select Variables', the 'Barcode Value' is set to '(!barcode)'. The 'Revisited Screen Values' section is collapsed. At the bottom right, there are 'Cancel' and 'Done' buttons.

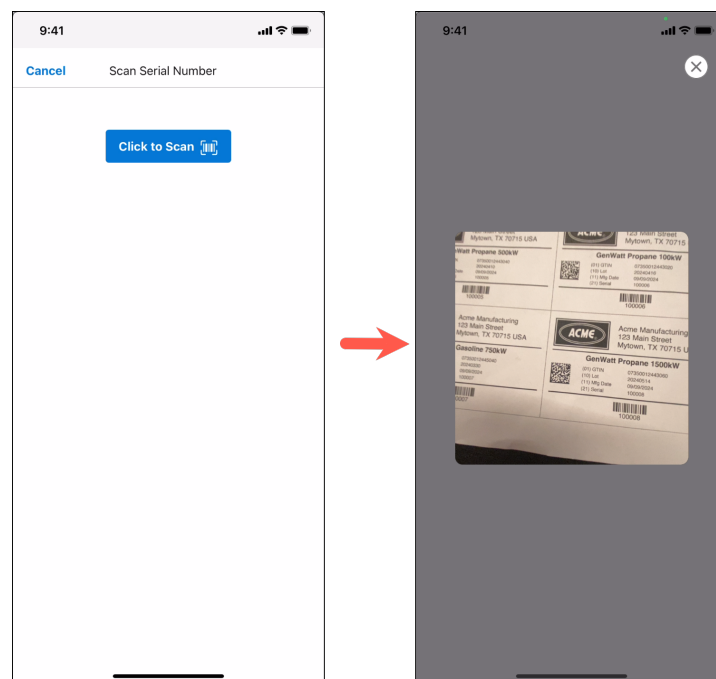
After completing steps 1-3:

- Save and activate the new Flow.
- Create an Action button on the Asset object, to launch the Flow.
- Add the Action button to the page layout for the Asset object.

When the flow is launched in the desktop environment, the user will be prompted to scan a barcode using a peripheral (e.g. USB, Bluetooth) scanner that is connected to the computer.



When the same flow is launched in the mobile environment, the user will be prompted to scan a barcode using the camera of the mobile device.



Implementation Assistance and Support

The support team at Gimbal Logic is equipped to help you get up and running with barcode functionality in Salesforce. Contact us at barcode@gimballogic.com