



**Building
Better
Businesses**

3B Forms – Quick Start Guide

This guide is intended to help you get up and running with 3B Forms as quickly as possible, and assumes you have completed all necessary package installation, configuration and setup as detailed in other supporting documentation.

[Working with 3B Forms](#)

[3B Forms Data Schema](#)

[Creating a Form record](#)

[Understanding the Form record page](#)

[Understanding Form Settings](#)

[Building a Form](#)

[Setting a Contextual Object](#)

[Adding Simple Questions](#)

[Understanding Question Element Settings](#)

[Adding Text and Merges to Forms](#)

[Adding Conditional Logic to Questions/Text](#)

[Adding Default Values and Data Validation to Questions](#)

[For default values:](#)

[For data validation:](#)

[Adding 'Complex' Questions](#)

[More information on working with Child Object Question Types](#)

[Adding Completion Actions to the Form](#)

[Styling a Form](#)

[Publishing a Form](#)

[Duplicating/Cloning a Form](#)

For additional support and guidance, please refer to the 3B Knowledge Base [here](#), or send an email to help@3b4sf.com.



Working with 3B Forms

3B Forms Data Schema

Objects:

- **Forms** - this is the record where you build each Form
- **Form Responses** - every interaction with a Form (via a link) will create a Form Response.

The Form Response record can be used to trigger automations and has data points that may be useful, such as:

- **Type;**
 - DRAFT - the Form was not completed.
 - COMPLETE - the Form was completed and updated Salesforce.
 - FAILED - the Form was completed and did not update Salesforce. The 'Error Message' will provide audit details as to the reason for this failure.
- **Main Record ID;** the Id of the record the Form used when updating Salesforce.
- **Form Live Duration;** how long from opening the Form to submission.
- **IPAddress;** of who submitted the Form.

Creating a Form record

To begin your form building journey, you will need to create a new Form record. You can do this either by:

- navigating to the '3B Forms' app in Salesforce and selecting the 'Forms' tab.
- or searching for and selecting the 'Forms' object in the Salesforce App menu.

When you are viewing the Forms object:

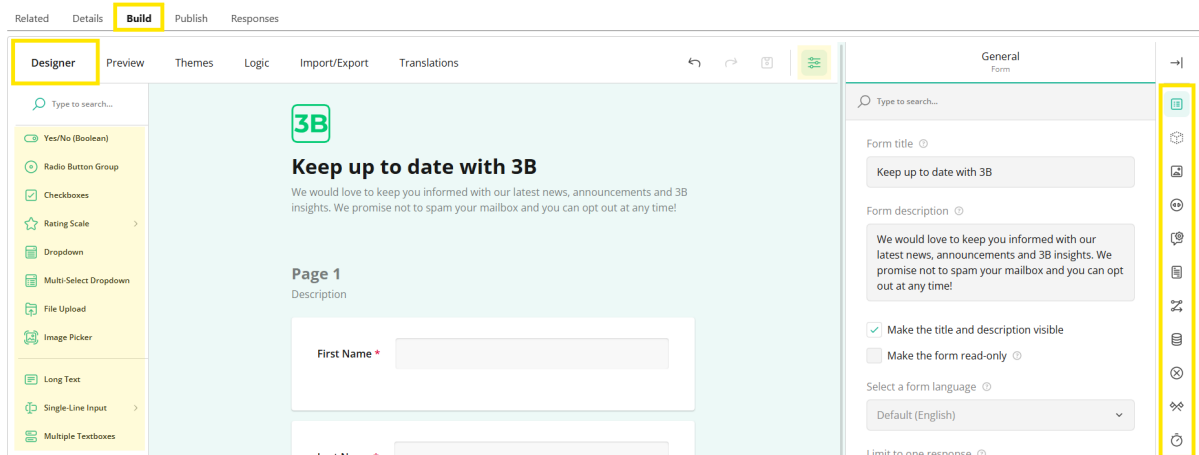
- Select the 'New' button to create a new form record.
- Simply give the form a name (this is for internal purposes only).
- Click 'Save'.

Please note you will need to create a new Form record for each form you wish to distribute, as these act as templates for your data collection needs - so please name them something meaningful so you can recall them as needed in the future.

Once you have created a Form record, select the 'Build' tab to access 3B's drag-and-drop form builder view.



Understanding the Form record page



In the 'Builder' tab, you will see 3 main sections:

1. **'Question Types'** panel (left) shows the available questions/elements you can add.
2. **'Drag-and-drop Canvas'** (middle) where you build the Form'.
3. **'Settings Panel'** (right) to control settings - this will automatically change based on which element in the canvas is selected and is shown at the top of this panel.

Understanding Form Settings

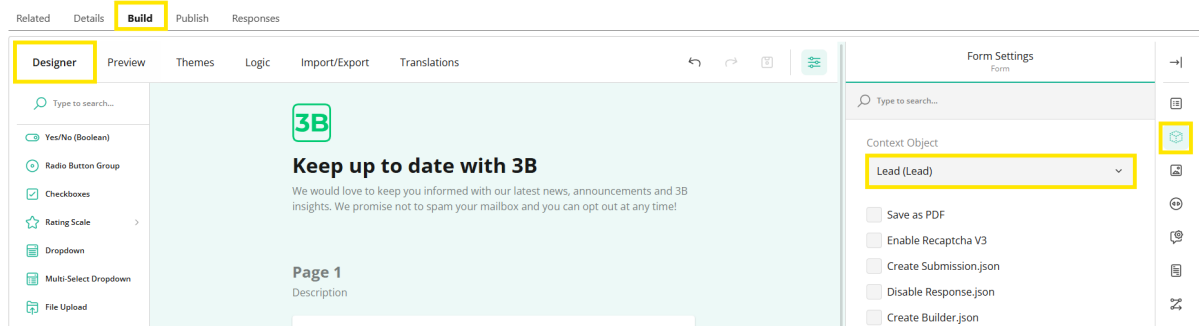
When you click in a blank space in the canvas, the 'Form Settings' will be enabled and shown at the top of the 'Settings Panel'. The settings available for the whole form and allow you to control:

- General** - the form's title, description and how the form responds.
- Form Settings** - which Salesforce object will be used as the context of the Form (3B is compatible with all standard and custom Salesforce objects).
- Logo in the Form Header** - the logo, size. and placement.
- Navigation** - navigation button labels and how the user interacts with the Form.
- Question Settings** - the position and order of question labels/descriptions, which can be over-ridden on a per question element basis.
- Pages** - the page order and page title/number visibility.
- Conditions** - variables and triggers used by the Form.
- Data** - how the form manages hidden data and progress.
- Validation** - how the form validates inputted data.
- "Thank You" Page** - what happens when the form is completed i.e. redirect to a URL or show a message, both of which can be dynamic based on how the form was completed.
- Quiz Mode** - whether the form has a time limit to be completed.



Building a Form

Setting a Contextual Object



Every Form needs a contextual object, which is the Salesforce object the Form will use to pull field values from, and push data back to.

- Access the Forms 'Settings'.
- Select 'Form Settings'.
- Choose the relevant Salesforce object you wish to use for your Form - this will control where you can map data from and to.

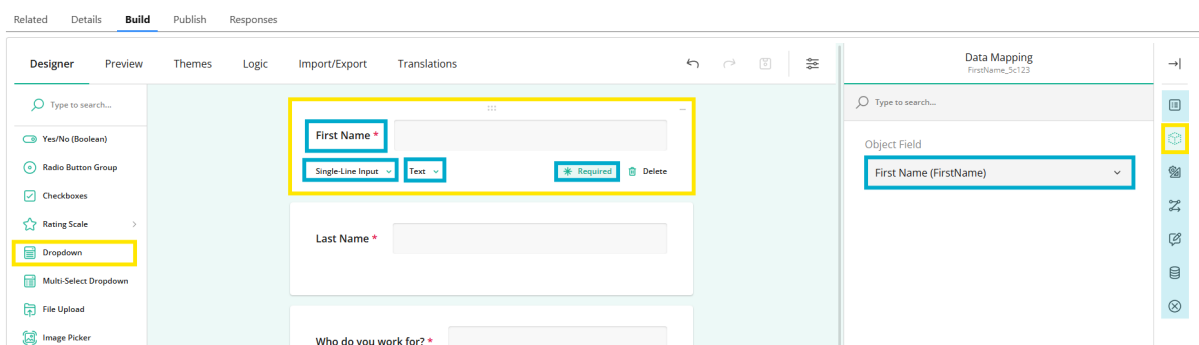
Example Form contexts:

1. You wish to create a lead generation form for your website that will create new Lead records - the context of this Form should be Lead.
2. You wish to create a personal details collection form, for existing Contacts to view and update their information - the context of this Form should be Contact.

Now you have set the contextual object of the Form, you can start adding questions.

Remember to 'Save' your Form at the top of the page to commit your progress.

Adding Simple Questions



Select the required question type from the 'Question Types' panel, drag-and-drop onto the canvas as required. You can reorder question elements or place them side by side, simply by dragging them to where you would like them to be placed in the canvas.



You can map a question to a Salesforce field by ensuring the question element is selected in the canvas, then select the 'Data Mapping' option in the settings panel - choose the relevant Salesforce field.

Please note:

- Not every question needs to be mapped to a Salesforce field, but it is important to note that if not mapped, how the question was answered will only be seen in a Form Response ([see Forms Data Schema for more information](#)), which requires a 3B licence.
- Formula fields can not be mapped to a question since you are not able to update the value of a formula field in Salesforce.
- If you are mapping Yes/No (Boolean) fields, please refer to additional configuration detailed in section [For default values](#):
- If a Salesforce multi-select/picklist is selected, by default 3B Forms will pull in the picklist values as choices. 3B Forms will honour the security settings of the selected field i.e. 'Restrict picklist to the values defined in the value set' ([more info](#)).
- When the 'File Upload' question type is used, it will store the file on the contextual object, unless used with the 'Parent/Child Object' question type.
- Not all question types can be mapped to Salesforce i.e. Image Picker.

Understanding Question Element Settings

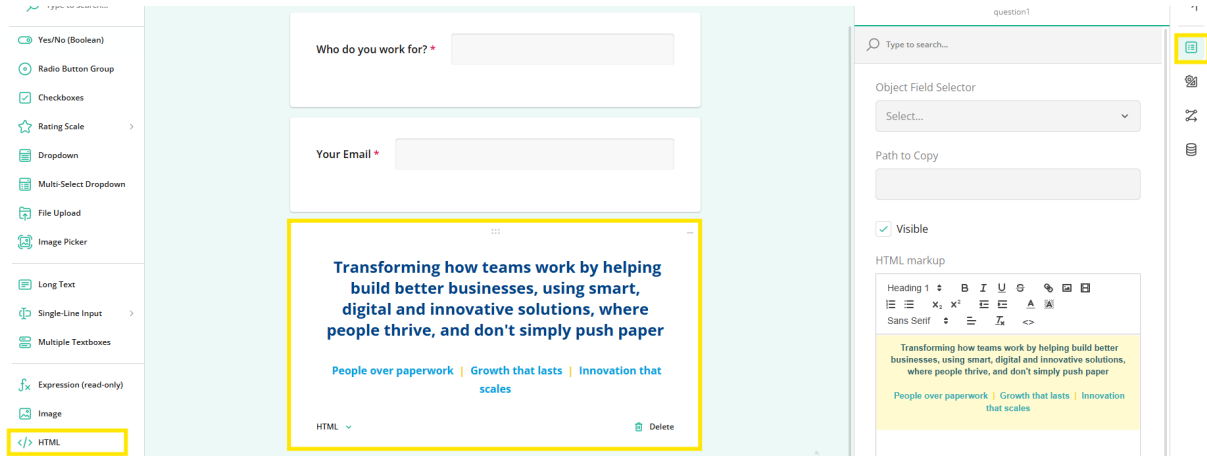
Select a question element in the canvas - the 'Settings Panel' is updated to show the settings for that selected item. From these settings you can control:

-  **General** - the title, description and whether the question is visible, required, read-only, or populated. For 'Single-Line Input' question elements, you can also control the type of data expected i.e. text, number, email.
-  **Data Mapping** - which Salesforce field on the selected object to map data from/to.
-  **Choice Options** (only available for Radio Button Group, Checkboxes, Dropdown, Multi-Select Dropdown, Rating Values, Image Picker) - the values available. If mapped to a Salesforce multi-select/picklist field, the Salesforce picklist values are automatically added. You can also add conditions to each choice value to determine what rules should be met as to whether an option is shown/hidden etc.
-  **Layout** - the position of question labels/descriptions, along with height/width.
-  **Conditions** - rules that make the question visible, required, pre-populated etc.
-  **Data** - default answers, and how Yes/No (Boolean) question types are handled.
-  **Validation** - whether responses should be entered in a set format i.e. using Regex.

Many of these settings can also be managed directly within the canvas by updating text as required, or by using the various question element toggles to turn things on and off, or set expected data type for 'Single-Line Input' question elements.



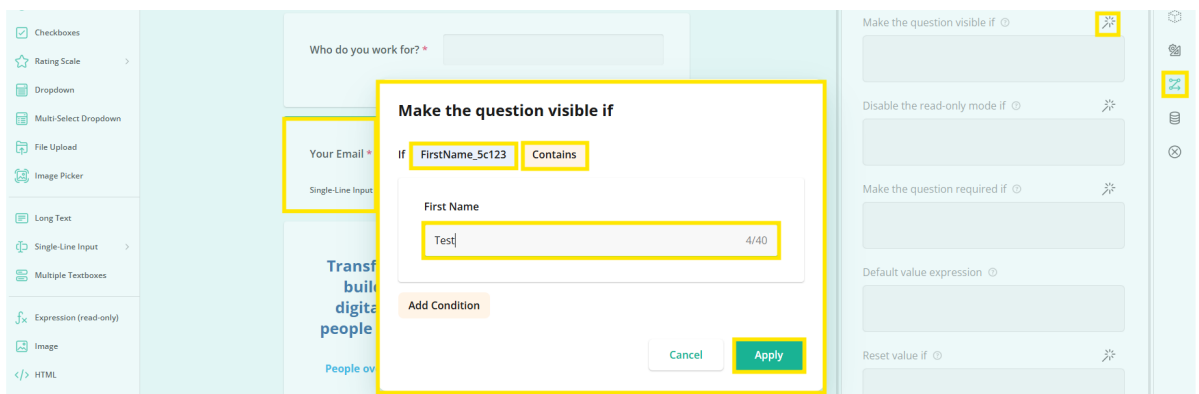
Adding Text and Merges to Forms



The 'HTML' question type can be added to Forms to provide instructional text or guidance. Text is entered using the question elements 'General' settings. You can also merge data from Salesforce by using the syntax `{{record.Field_API_Name}}`.

This can also be used in the Form title/description, question title/description, and also for default values – but can only be used to merge data from the contextual object.

Adding Conditional Logic to Questions/Text



- Select the question element in the canvas.
- From the 'Settings Panel' select 'Conditions'.
- There are various conditions that can be added using the 'wand' symbol next to the required rule logic. Use the pop-up box and point-and-click to build your logic – please note you are able to apply and/or logic, but not able to apply advanced customised logic such as '1 and 2 and (3 or 4)'.
- Click apply to accept this rule.

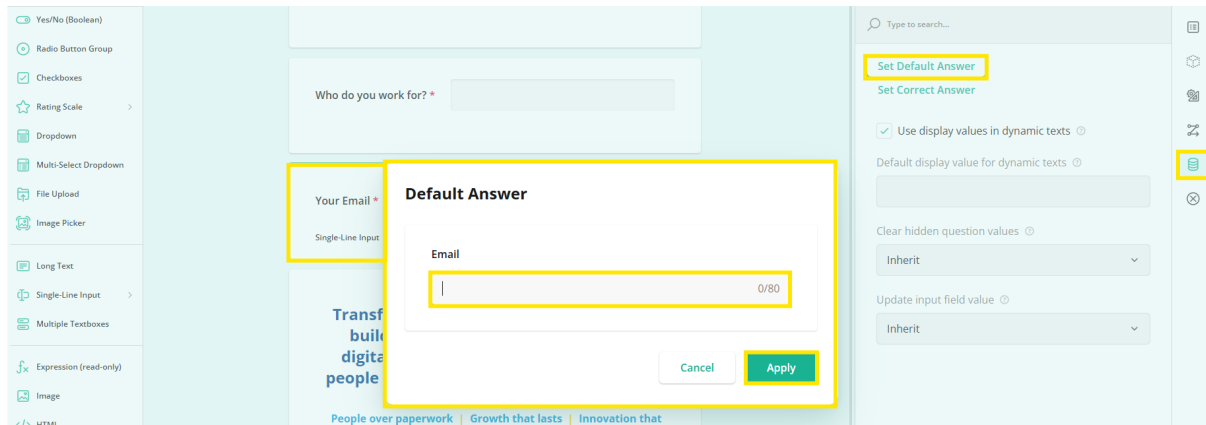
You can then copy and paste the code to other conditions if required.

All rules are rolled up into the 'Logic' tab at the top of the page, so you can review and manage all rules from a single place if required.

Page rules work in a similar manner, however are accessed by selecting the page, then using the 'Settings Panel' for the page and accessing the 'Conditions' settings.



Adding Default Values and Data Validation to Questions



For default values:

- Select the question element in the canvas.
- From the 'Settings Panel' select 'Data'.
- Use the 'Set Default Answer' to input default values as required – you can use the merge syntax `{{record.Field_API_Name}}` to pull in Salesforce data.

Please note that when using the 'Yes/No (Boolean)' question type, the 'Data Settings' can also be used to determine how the selected response is mapped back to Salesforce. As an example, if the field is mapped to a Checkbox in Salesforce, you will need to update:

- 'Value to store when "True" is selected' to **true**
- 'Value to store when "False" is selected' to **false**

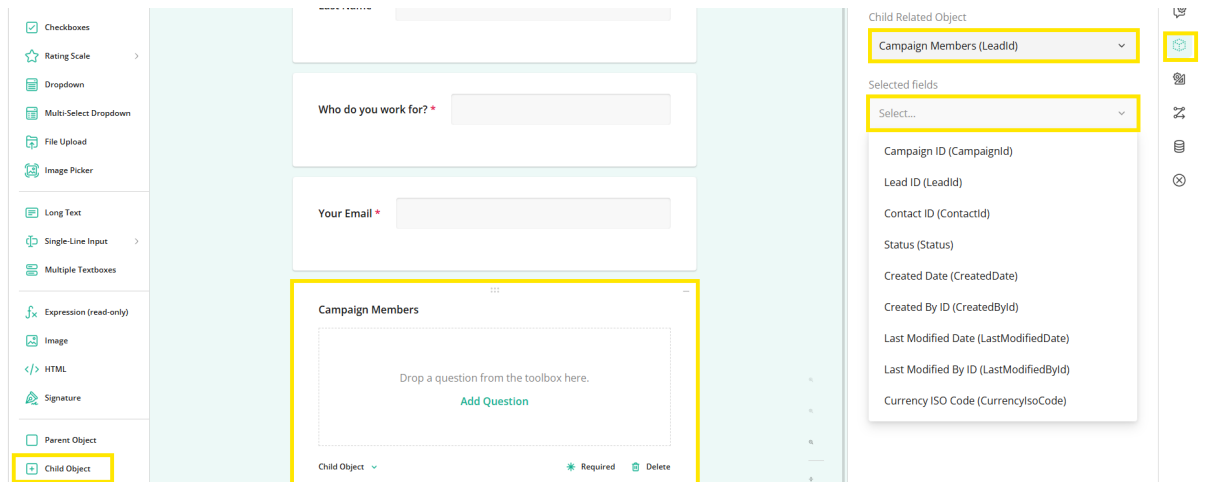
This is because the Form reads responses as Yes/No, which is not the same as true/false in Salesforce checkboxes.

For data validation:

- Select the question element in the canvas.
- From the 'Settings Panel' select 'Validation'.
- Click 'Add new rule' and follow prompts as required to either insert an Expression, Number, Text, Regex, or Email validator.



Adding 'Complex' Questions



The vast majority of question type elements will be straightforward and used to simply pull data from Salesforce, and map data back as required. There may be use cases where you wish to allow the user to interact with both Parent and Child objects of the Forms contextual object. You can do this by using the 'Parent Object' and 'Child Object' question type elements. Both work in a similar way:

- Drag the 'Parent Object' or 'Child Object' question element into the canvas.
- With the question element selected, from the 'Settings Panel' choose 'Data Mapping' – this will allow you to select the parent/child object related to the contextual object.

Once the parent/child relationship has been defined, you can add question elements in the same way as with the 'Simple Questions', or by using the 'Selected Fields' option in the 'Data Mapping' Settings– please ensure that if you drag-and-drop question types onto the canvas, they are held within the Parent/Child Object element.

If adding a 'File Upload' question type to a Parent/Child Object element, the file will be stored on the respective Parent/Child record.

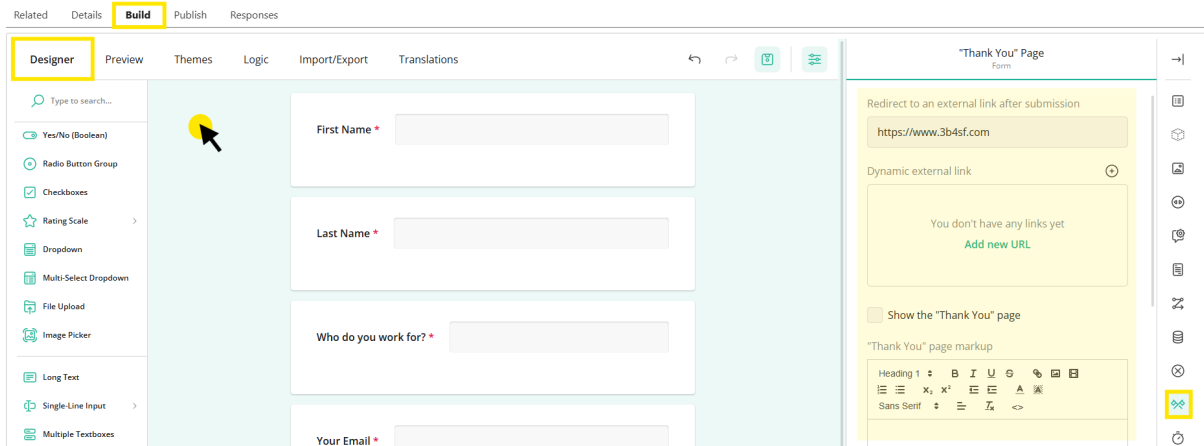
More information on working with Child Object Question Types

You can use the 'Data Mapping' Settings to add a 'Custom SOQL Filter' – this helps control the number, order and which child records are pulled into the Form, such as `Id = null ORDER BY Start_Date__c ASC`.

You can use the 'General' Settings to determine how child records are displayed; as a vertical list, or horizontally via a carousel/tabs. You can also control whether the user has the ability to add/remove child records from the Form. Please note that if the remove option is enabled, this will not delete child records from Salesforce – only the Form.



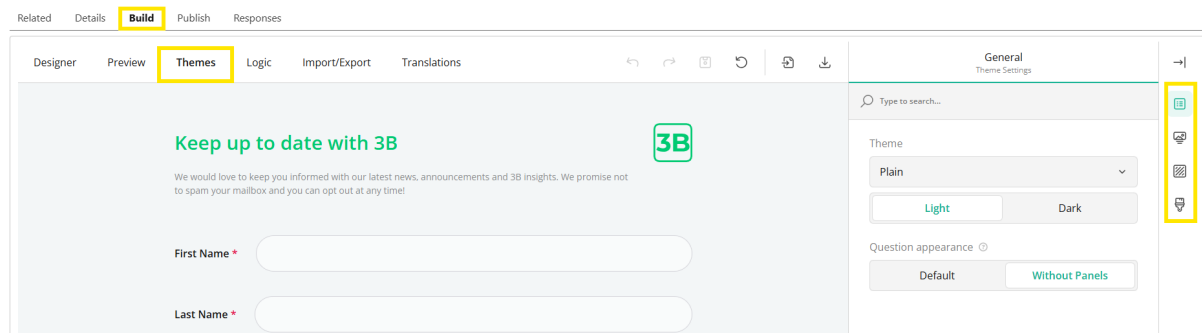
Adding Completion Actions to the Form



- Select the canvas to change settings to the context of the Form
- Select "Thank You Page" Settings
- Update completion actions as required either by:
 - entering a "Thank You" page message - or extending this to be dynamic based on how the Form was submitted.
 - populating the 'Redirect to an external link after submission' field with a URL - you can also use the merge syntax `{{record_Field_API_Name}}` to include variables from the contextual record.
 - adding 'Dynamic external links' - this works in the same way as conditional logic.



Styling a Form



In the 'Build' tab, select the sub-tab 'Themes'. The settings on the right allow you to update the:

-  **General** – overall look and feel of the Form
-  **Header** – background colour, logo/title placements etc, font size and colour
-  **Background** – colour or images
-  **Appearance** – accent colours, or even use the 'Advanced mode' to make updates to every element within the Form.

Select each setting and make changes as required.

Remember to 'Save' the theme at the top of the Form to commit your progress – please note the Theme and Form save buttons work independently of each other.



Publishing a Form

Related Details Build **Publish** Responses

Did you know
You can embed this form on a website, send it as an email link, embed it as a quick action and use it in communities. Additionally, non-salesforce users can interact with the form without any additional licenses if you set a Guest Site

Non-secure link https://3bforms.my.salesforce-sites.com/3845F/apex/b3f__Form?id=a7b08000003d6rd6A&

Secure link https://3bforms.my.salesforce-sites.com/3845F/apex/b3f__Form?key=48p8vMOCCquaj3GzCL548APX4N%2FL7ocVQOW3NvgffhZ%2Fb%2FKq9B919xbnsTQGIQW56ddM7OYmSLagF1s8eMuk44eV%2F4fkoDhvtKUUe%282Oy11NyzzsRLz0Ml

Embedding Code
`<script src="https://3bforms.my.salesforce-sites.com/3845F/resource/b3f__FormCompleteWC/index.js" defer="true" type="module"> </script> <script src="https://3bforms-dev-ed.my.salesforce.com" site-url="https://3bforms-dev-ed.my.salesforce-sites.com" ai-endpoint="https://ai.3bapi.com/ai" form-id="a7b08000003d6rd6A" record-id="" version-id="" key="" ai-model-name="gpt-4o-mini" captcha-site-key="6Ld7S8hgAAAAAAS2NOCB3yYDVxewotLjMhZG91dGVy" org-id="SET_THIS_FOR_AI" user-id="SET_THIS_FOR_AI" page-ip-address="YOU_MUST_SET_THIS" session-id="NULL_SESSION_ID" user-type="Guest" theme-type="light"> </script>`

* Site Select X3B4SF X

* Object Select Lead (00Q) X

Record Search Records... Q

Version Select

+ New Param

- Click the 'Publish' tab.
- Select a 'Site' – these are the Force.com or Digital Experience Sites setup as part of initial installation, configuration and setup.

This creates a Form URL in the format

[SiteURL]/apex/b3f__Form?id=[Form_Record_Id]

This link can be used which will create new records in Salesforce. To make Forms specific to a record in Salesforce:

- Select the Forms contextual object from 'Object Select'
- Select 'Record'

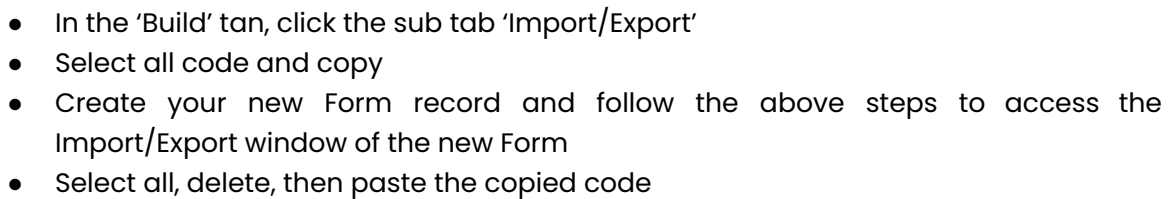
This changes the Form URL to:

[SiteURL]/apex/b3f__Form?id=[Form_RecordId]&recordId=[Record.Id]

This URL can be used dynamically in emails etc to send a link to a particular individual, where the [Record.Id] variable is replaced with the relevant Salesforce Record Id of the record that you wish the Form to access/update.

In addition:

- You can also use the 'Embedding Code' to add the Form to websites etc.
- You can add additional parameters to the URL to pre-populate question elements as required – simply use the '+ New Param' box to select Salesforce fields and their respective variables. You will see the parameters added to the Forms URL.



3B Forms – Quick Start Guide V1.0 Sept 2025