

How to Use Calculations in your Digital Form Template

Everything you need to help you get started with GoFormz Calculations!

proservices@goformz.com

sales@goformz.com

Table of Contents

Introduction: The value of digital forms

What are Calculations?

How to utilize Calculations within your digital form Template

How to add Calculations to your digital form Template

Use Case Examples:

Automatically complete mathematical calculations

Dynamic Field Properties

Instantly referencing Database information

Auto-Populating user information

Popular Calculations

Additional Resources



Introduction: The value of digital forms

Digital form solutions have evolved into one of the most powerful tools of the modern workplace. The amount of time and energy spent on daily tasks can be significantly reduced by implementing digital solutions, which boost productivity and efficiency across your business operations.

Whether you're interested in eliminating paper for environmental reasons or simply looking to streamline daily processes and enhance business workflows – digital form solutions like GoFormz make your work-life easier by:

- Transforming tedious manual processes into automated, data-oriented workflows
- Creating a constant stream of rich, high-quality data throughout your organization
- Eliminating data gaps between your digital systems
- Producing actionable insights to make informed, effective business decisions
- Eliminating data collection obstacles, resulting in significant time and cost savings...and so much more

Digitizing your paper forms empowers businesses to streamline their business operations, and unlock powerful efficiency improvements unattainable when using paper forms and rigid PDF fillers. GoFormz holds multiple patents on its digitization technology, making it the only platform that preserves the original look and feel of your existing paper forms – allowing your operation to maintain branding, streamline user adoption, and preserve specific formatting requirements (handy when working with government agencies or compliance activities).

Requiring no coding experience or IT support, GoFormz digital forms can be easily configured in an online, drag-and-drop form builder. These digital forms can even be equipped to collect a variety of new data types, like *Signatures, Images, Barcodes,* and *Maps,* and can include easy-to-use form *Logic.* GoFormz also provides an additional viewing mode, *List View,* allowing form creators the flexibility to create forms entirely from scratch – no form required!

And the power of digital forms doesn't end there! Digital forms can even be filled out while working offline in remote locations – eliminating costly delays and accelerating the flow of information throughout your operation. Once forms are completed, they are automatically stored in the GoFormz Cloud as well as routed to collaborating team members, clients, integrated platforms, and more.

Among these valuable digital form features are *Automatic Calculations*, one of the most powerful benefits of digitizing your data collection and documentation. For GoFormz users, this often means implementing *Calculations* to guide data entry, improve the accuracy of calculations and entered data, and automate field actions. In this eBook, we'll explore what *Calculations* are, the value they bring, and the most popular *Calculations* leveraged by our users.

The Calculation Builder



Calculations enable your form fields to perform anything from simple addition or multiplication to complex *Conditional Logic* – directly within your form. In other words, they perform actions based on data entered into a specific form field.

These Calculations are configured within the Template Editor, using the Calculation Builder. The Calculation Builder, which can be accessed within the Properties Panel, guides form builders through the addition of calculations. Within the Calculation Builder is an index of all available functions, fields, and variables GoFormz users can leverage within their form Template. As a user types a formula into the Calculation Builder's Formula Bar, the list of available functions, fields, and variables truncates to only display those relevant to the entered formula – simplifying and streamlining the formula creation process.

The Calculation Builder also provides inline validation and error validation while a formula is being entered – highlighting problematic segments of the formula that require correcting. Once a formula has been created, users can easily test the formula, allowing users the opportunity to easily troubleshoot any potential issues with their Calculations.

How to utilize Calculations within your digital form Template

Calculations allow Template builders to easily add simple, yet powerful, logic to their digital forms, including:

- Default Values: A value that automatically appears in a field when a form is generated.
- Dynamic Field Properties: The Required, Background Color, and Visibility properties of your form fields can dynamically change, based on field selections, input data, Group membership, and more.
- Database filtering: Present a limited set of options to a user based on the value of another field.
- Default form names: Instantly populate a form name, which can include user details, information entered into your template, and more.
- Formatting properties: Standardize how data is displayed once entered into your form (e.g. time, date, or text formatting).

How to add Calculations to your _ digital form Template

04

You can equip your form *Template* with these calculations using the *Calculation Builder*. The *Calculation Builder* is easily located by accessing the *Template Editor*:

- 1. From the homepage, navigate to the *'Templates'* tab and select which *Template* you would like to edit. Once you select your *Template*, the GoFormz *Template Editor* will load.
- 2. Select the field you would like to add a *Calculation* to and navigate to the *Properties* panel (to the right of your *Template*).
- 3. Locate and select the *Default Value* property and the *Calculation Builder* will load.
- 4. Within the *Calculation Builder* you can enter and test your *Calculations*. To get started, simply type a '=' and options will begin to populate.

To better understand how users can add and utilize *Calculations* within your form *Templates*, let's explore four popular *Calculations* you can add to your *Templates* today.

Template Fields Add New Fields Paid Step 1 & 2:	
Add New Fields Field Field Dated View	
Step 1 & 2:	
DICD TOLE.	
Drag and drop fields	
from the Template	
Fields Panel to enable Step	3 & 4:
the Properties Panel	into the Default
to display! Value	Property and
the C	Calculation
Builde	er will populate!

05 Calculation Examples

Use Case One: Automatically complete mathematical calculations

Mathematical formulas are being used by thousands of GoFormz users to simplify and expedite the completion of calculations while enforcing highly accurate data capture.

From tracking wages and hours within timesheets to providing estimated costs within quotes – you can easily automate the completion of common calculations using simple formulas.

01

Use Case Example: Determining the total of a table column

Calculations are frequently used within *Table* fields. In this example, we'll use *Calculations* to find the total sum of a column in a *Table* field.

To get started, we'll first identify what elements need to be included in our formula. GoFormz calculations reference *Table* cells as [column][row]. Depending on the complexity of the calculation, you may also need to define the specific *Table* within the calculation string, for example, if you needed to multiply a cell in one *Table* with a cell in another *Table*. *Tables* are defined as "[Table Name]!".

For this example, the *Table* will include 4 columns and 4 rows. With the *Table* field selected, you will notice an additional grey column to the left of the *Table* and a grey row above. When selected, the *Properties* panel will reflect details for that particular row or column of the *Table*.

Example Table:	ltems	Quantity	Price	Totals
Column 1: Item	 ltem	Quantity	Unit Price	Total
Column 2: Quantity	 ltem	Quantity	Unit Price	Total
Column 3: Unit Price	 ltem	Quantity	Unit Price	Total
Column 4: Total	 ltem	Quantity	Unit Price	Total
	 	[Subtotal	

To get started, open the *Properties Panel* for the "Totals" column by selecting the grey column header. Locate the *Default Value* section – this is where you will add your formulas. Select "Use Individual Values" from the drop down, and additional rows will populate, allowing you to now set up your four-row equation.

=IF([2] [1] * [3] [1] > 0, [2] [1] * [3] [1], "")								
Logic Functions If And Or	If Returns one value if a condition is Syntax	true and another value if it's false.						
Not False True	If(condition, trueValue, falseValue)							
Math Functions Sum Average	Formula =If(2 = 2, "first", "second")	Result first						
Min =If("false", "first", "second") second Press Enter to finish editing Test Formula ? Learn model								

For this equation, we'll use the following syntax: [Quantity][1]:[Unit Price][1]

Following the same naming conventions mentioned above, the first row, Row 1, would look like this:

=IF([2][1]*[3][1]>0, [2][1]*[3][1],"")

Now follow the same procedure for your remaining three rows, adjusting the number to reflect each specific row:

Row 2: =IF([2][2]*[3][2]>0, [2][2]*[3][2],"") Row 3: =IF([2][3]*[3][3]>0, [2][3]*[3][3],"") Row 4: =IF([2][4]*[3][4]>0, [2][4]*[3][4],"")

Now that we have our *Table* set up to calculate the total of each row, let's calculate the grand total of the "Total Column". In order to do this, set up a regular *Text* field below your table and access the *Calculation Builder* for the *Default Value* property within the *Properties Panel*. Now, add the following calculation:

=SUM([Example Table]![4][1]:[4][4])

This equation instructs your *Table* to sum the values in Column 4, from Rows 1 to 4 in our example *Table* (don't forget to name your *Table* – you'll need to specify your *Table* name in your formula).



Use Case Example: Instantly apply sales tax and discounts

For this example, we will build off of our previous example and apply sales tax and discounts to our final total. First, let's create a few new *Text* fields and title them: "Discount Rate", "Discount", "Adjusted", "Tax Rate", "Tax", and "Grand Total".

First, we will configure our "Discount Rate", which will automatically apply a discount to the total amount when a value is entered into the "Discount Rate" field. For this example, we will use a simple Text field, but this functionality can also be applied to *Drop Down* menus pull from a *Database*.

ltems		Quantity		Price	Totals	
ltem		Quantity		Unit Price	Total	
ltem		Quantity		Unit Price	Total	
ltem Q		Quantity	,	Unit Price	Total	
ltem		Quantity		Unit Price	Total	
Discount Rate	Discount Rate			Subtotal	Subtotal #	
				Adjusted	Adjusted	
Discount	Discount		1	Тах	Тах	
Tax Rate	Tax Rate			Grand Total	Total Cost	

=IF([Subtotal #] * [Discount Rate] > 0, [Subtotal #] * [Discount Rate], "") Done						
Fields	Subtatal #					
Subtotal #	Field Type: Text					
	Default Value: = SUM([Example Table]![4][1]:[4][4])					
	Syntax					
	[Subtotal #]					
	Example					
	Formula	Result				
	= [Subtotal #]	some value				
Press Enter to finish editing Test Formula 👔 Learn						
First formula listed below and the syntax you should see when you click on the field property [Subtotal #]						

To get started, select your "Discount Rate" *Text* field. This field will represent the discount percentage that will be applied to the subtotal. Next, select the "Discount" field, navigate to the *Default Value* in the *Properties Panel*, and add formula below:

=IF([Subtotal #] * [Discount Rate] >0, [Subtotal #] * [Discount Rate],"")

Next, we'll configure our "Adjusted" field to automatically apply the discount calculated above, by adding the following formula to the *Default Value*:

```
=IF([Subtotal #] - [Discount] >0, [Subtotal #] - [Discount],"")
```

Now, you have successfully applied a discount to your subtotal!

Next, we will apply the tax. We will do so by selecting the "Tax" *Text* field (however, using a *Drop Down* menu to pull from a *Database* of tax codes would be the best practice). This *Text* field will represent the tax value we will apply to our discounted total. (When applying tax remember to use decimals e.g. California Sales tax is 7.25% which means .0725 would be the value you would input into the *Text* field).

=IF(SUM ([Adjuste	ed] , [Tax]) > 0, SUM([Adjusted] , [Tax]), "") Done			
Fields	Adjusted				
Adjusted	Field Type: Text				
	Default Value: = IF([Subtotal #] - [Discount] >0, [Subtotal #] - [Discount], "")				
	Syntax				
	[Adjusted]				
	Example				
	Formula	Result			
	= [Adjusted]	some value			
Press Enter to finis	sh editing	Test Formula (?) Learn more			
Final formula to add to the <i>Default Property</i> within the "Grand Total" <i>Text</i> field. This illustration displays what you would see if you selected "Adjusted".					

With the "Tax" Text field selected, locate the *Default Value* and add the following formula (adjusting as needed for your specific tax rate):

=IF([Adjusted] * [Tax Rate] >0, [Adjusted] * [Tax Rate],"")

Finally, we can sum all the values calculated above to produce a grand total in the "Grand Total" *Text* field. In the *Default Value* of this Text field, add the following formula:

=IF(SUM([Adjusted],[Tax]) >0, SUM([Adjusted],[Tax]),"")

Congratulations! You have successfully configured your *Table* field to automatically apply discounts and sales tax. Leveraging *Automatic Calculations*, your users can seamlessly calculate grand totals without having to waste time on manual math – accelerating form completion, eliminating opportunities for human error, and streamlining your billing, quoting, and purchasing processes.

Use Case Two: Dynamic Field Properties

Dynamic Field Properties (DFPs) allow your form to automatically modify and manipulate the look and behavior of fields, based on field selections, entered data, and a variety of other factors. DFPs are used to guide data capture, highlight missing information, instantly provide additional fields, hide fields from Users and Groups, and much more – providing a more flexible and intuitive data entry experience.

For example, you could use a *Calculation* in conjunction with the *Required* property of a *Signature* field, instructing the field to become *Required* if a user meets certain criteria. This ensures critical fields are completed before a form is submitted.

Form fields can even be seamlessly tailored to specific *Users* and *Groups* – customizing the user experience and removing unnecessary data entry. For example, a form could be configured to only display a section of fields if a user is a member of a 'Supervisors' Group.

For this section, we will explore two common *Dynamic Field Properties* use cases popular amongst our users.

Use Case Example: Make a field Visible if a user selects a specific checkbox

Leveraging Dynamic Field Properties to make specific fields Hidden/Visible is a powerful method for guiding user data entry. For this example, we will make a *Text* field Visible when a user selects a specific Checkbox field.

This use case is a powerful example of how *Calculations* can be leveraged to tailor digital form fields to a user's specific needs.

First, drag a *Checkbox* field onto your *Template*. Next, drag a *Text* field and place it directly next to the *Checkbox* field. With the *Text* field still selected, navigate to the *Properties Panel* and locate the *Visible* property. Select the [...] icon to open the *Calculation Builder*. Now add the following *Calculation* (remember [Checkbox] is the name of the checkbox we are pointing to, so make sure to adjust your *Calculation* to use the name of your *Checkbox* field):



Now if you preview your form and select the *Checkbox* field, you will now see an additional *Text* field appear. If you wish to highlight this newly visible field, you can dynamically change the field's *Background Color* by adding the following equation to the *Background* property:

```
=IF([Checkbox]="true", "#FF2400", "#D8F5FF")
```

Now that we have discovered how using *Calculations* with *Dynamic Field Properties* can guide users through a form, let's look use this next example to explore how *Dynamic Field Properties* can enforce data capture rules and ensure critical form fields are completed before a form can be submitted.

	=IF(IsInGroup("(098f311e-a398-4c0e-9697-a1ea0135 _	5b204"), "true", "false")	Done
02	Groups Administrators Supervisors Contractors Logic Punctions If And Or Not False True Math Functions	IsInGroup Returns true if the current user is a member Syntax IsinGroup[groupId, [groupId],) Example Formula =IsinGroup[*098/311e-a398-4c0e-9697- a1ea0135b204) Groups	r of any specified groups.	
	Press Enter to finish	editing	Test Formula	Learn more

Use Case Example Two: Dynamically make a field Required if a User is a member of a specific Group

Required fields make it easy to enforce the capture of critical information, like Signatures. The Required property can also be made dynamic, allowing form builders to easily tailor their digital documents to the unique data capture needs of the user. This is made possible by using Dynamic Field Properties and the ISINGROUP function.

The ISINGROUP syntax references the user's *Group* membership. This functionality is incredibly useful when managing a variety of teams or departments with varying responsibilities, permissions, and job needs. The ISINGROUP function can be applied to a variety of field properties to dynamically change how they display or operate based on the user's *Group* membership.

To get started with this simple formula, start by selecting the field you wish to make dynamically *Required* and navigate to the *Properties* panel. Select the [...] next to the *Required* property, to open the *Calculation Builder* and enter your a calculation:

=IF(IsInGroup("Group Number Here"), "true", "false")

In order for this formula to work properly, you will need to include the Group ID number in your formula. There are three simple steps when locating a Group ID:

- 1. In the GoFormz Web App, navigate to Users & Groups
- 2. Select the appropriate Group
- 3. The Group ID will be located to the right of your Group Name in the URL

Now, when members of a specific group generate a form, they must complete the *Required* fields before their document can be submitted – improving data accuracy and completeness without disrupting the data entry of other groups.

Use Case Three: Instantly referencing Database information

GoFormz can make filling out forms a breeze by pulling structured information from various *Data Sources* to automatically populate multiple fields at once. This is accomplished through the use of *Database* fields, which pull information from your GoFormz data or data collections stored within connected platforms – streamlining form completion processes and eliminating opportunities for human error.

Database fields can be used to auto-fill fields in your form based on field selections and entered data. For instance, choosing a customer's name can automatically fill relevant fields with that customer's phone number, address, email, and more. This leads to tremendous time savings, improved data accuracy, and a more streamlined data collection process.

01

Use Case Example: Scan Barcodes to instantly populate corresponding fields

Barcode Scanning can be used to significantly streamline data entry, especially when completing documents like inventory forms and equipment inspections. Leveraging *Calculations* in conjunction with *Barcode Scanning* to auto-populate corresponding fields significantly reduces human error, eliminates opportunities for missing data, and allows users to work quickly and efficiently.

Begin by dragging a *Barcode* field from the *Template Fields* panel onto your *Template* and adjust its size and location accordingly. Now locate and select the field you wish to populate with *Barcode* data. For this example, we'll use a *Text* and *Number* field where we would like product information to appear.

The syntax of the formula we will use for this example will be: =Database("dataSourceName", [key field], "columnHeader")

This syntax simply instructs the *Text* field to use the *Key Field* when referencing the database and populate with the corresponding value from the specified Column Header. For this example, we will auto-populate the product SKU, price, and name, using the following calculation strings:

```
Text Field (Name): =Database("Product List", [Product Code], "Name")
Text Field (SKU): =Database("Product List", [Product Code], "SKU")
Number Field (Price): =Database("Product List", [Product Code], "Price")
```

These formulas will be added to the *Default Value* property of each field, via the *Calculation Builder*. With these three fields configured using the calculations listed above, when a user fills out a form using this *Template* they will be able to scan the *Barcode* of a product and instantly populate its name, SKU, and price.

Use Case Example: Auto-populate customer information using Drop Down menus

A Database field works similarly to a Drop Down menu – it shows a list of items from which a user can make a selection, drawing this list of items from a GoFormz Data Source column. Leveraging Drop Down menus provide structure to entered data, limiting what information can be input to a predetermined set of items, and reducing opportunities for error.

Keep in mind that before adding this functionality to your *Template*, you must have a *Data Source* set up. In this example, we will use an "Example Data Source -Customers" *Data Source*. This *Data Source* lists the customer's name in the "Key" column, and stores information about each customer in columns called "Phone", "Address", "Email", and much more.

Work	c Orde	r		WORK ORDER #: Work Order DATE: Date		Type Name Customer Name Description	Dutabese
Company Name: Phone:	Customer Name Phone			Complete Complete	J	DistaSource Display Column Filter Allow Other Required Author Only Identifier	Example Data Source V Key V /k Yes V No V No V
Address:	Address					Default Value	ji ji
City:	City	State: State	ZIP: Zip			Allow Override	No ¥
PO #: Technician:	PO # Technician					Display Align Fort	Left v Verdata v



To begin, simply drag-and-drop your *Drop Down* menu onto your *Template*. Next, add *Text* fields for each of the items that you wish to auto-populate. For this example we will add *Text* fields labeled Phone, Address, City, State, and Zip. For each of these *Text* fields, set up the *Default Value* in the *Properties* panel to draw data from the appropriate *Data Source* column using the same syntax formula above:

=Database("dataSourceName", [key field], "columnHeader")

For our example, the *Default Value* for the "Phone" field is **=Database("Example Data Source - Customers", [Customer Name], "Phone").**

This tells GoFormz to reference the value selected for "Customer Name" within the "Example Data Source - Customers" database, and then populate the corresponding value from the database's "Phone" column in this field. Repeat this step for each of the fields you wish to automatically populate when a Drop Down item is selected, using the following formulas:

Address Field: =database("Example Data Source - Customers", [Customer Name], "Address") City Field: =database("Example Data Source - Customers", [Customer Name], "City") State Field: =database("Example Data Source - Customers", [Customer Name], "State") Zip Field: =database("Example Data Source - Customers", [Customer Name], "Zip")

Selecting a "Customer Name" will now auto-populate all the related fields, boosting data accuracy and saving time each and every time the form is used.

Use Case Four: Auto-Populating user information

GoFormz can instantly populate a user's information into form fields when they generate a new form. This capability not only accelerates form completion and improves data integrity but also improves accountability. For example, by leveraging *Location* fields to instantly populate a user's *GPS* coordinates, form builders can effortlessly ensure a user's precise location is entered as soon as they generate a new form.

Use Case Example: Instantly populate a user's contact information

You can enable your form to auto-fill current *User* information (like a *User's* name or contact information) when they create a new form, by adding *Calculations* to your form fields. To pre-populate *User* information in your forms, we will be adding 'current user' *Calculations* to the *Default Value* property of *Text* fields.

For this example, we will populate the user's 'First Name', 'Last Name', 'Email', and 'Phone Number' (you can also populate a users 'ID').

To get started, add *Text* fields for each of the values we mentioned above. Select the 'First Name' *Text* field and navigate to the *Default Value* field property in the *Properties* panel. The *Calculation Builder* will open. Enter 'Current User' and you will see five available calculations you can easily apply to your *Default Value*.

First Name: =CURRENTUSER_FIRSTNAME Last Name: =CURRENTUSER_LASTNAME Phone Number: =CURRENTUSER_PHONENUMBER Email: =CURRENTUSER_USERNAME

Now when a user generates a new form, their contact information will automatically populate within the correct fields. This information is pulled directly from their account information found within the settings of your account.



Use Case Example: Instantly populate a user's exact location

GPS fields improve transparency into your team's activities across job sites, franchise locations, offices, and more. These *GPS* fields can be instantly populated when a user generates a new form, reducing data entry demands and ensuring an accurate location is entered every time.

To automatically populate a user's location when they generate a new form, simply follow the instructions below:

Drag-and-drop a *Locations* field from the *Template Fields* section onto your *Template*.

With your *Location* field selected, navigate to the *Properties* panel and locate the *Display* property. This property allows you to determine if you would like your location field to appear as 'text' (*GPS* coordinates) or as a *Map*.

Next, locate the *Default Value* property and select 'Auto'. This instructs our form to automatically fill the *Location* field with either *GPS* coordinates or a *Map* displaying the user's location when a new form is generated.

Finally, Save and Publish your form! Now, whenever a user is completing a form, their exact location will automatically populate!

06 Popular Calculations

Calculations may feel daunting, but the opportunities are endless! In case you didn't see your specific use case mentioned above, or simply want to learn other ways to leverage *Automatic Calculations*, here are additional formulas commonly used by GoFormz users:

Expiration dates from time of purchase (this one is 30 days from the quote date – if you want a different range then adjust the "+30" variable) =If([QuoteDate]="","",DateValue([QuoteDate])+30)

Formula to find age =IF([Date of Birth]="","",Year(DateValue(NOW()) - DateValue([Date of Birth])) - 1900)

The calculation for making a *Checkbox Group Required* based on the *Checkbox Group* items. You can also swap the true and false to have the opposite effect. =IF(OR(FIND("Checkbox Group Item Name",[Checkbox Group Field Name]) > -1,FIND("Checkbox Group Item Name",[Checkbox Group Field Name]) > -1),"true","false")

A *Table* that will sum all of the items that have been checked off on a column (most commonly seen in order forms). Refer back to the Example One Use Case for the syntax formula reference.

=COUNTIF([Table 1]![1][1]:[1][5], "true")

Finding the difference between Start Time and End Time on timesheets (or warrant forms and job logs) with overnight functionality. =IF(OR([Time Start]= "",[Time End] = ""),"",IF(DATEVALUE([Time End]) < DATEVALUE([Time Start]), DATEVALUE([Time End]) - DATEVALUE([Time Start]) + 1, DATEVALUE([Time End]) - DATEVALUE([Time Start])))

07 Additional Resources

If you found this eBook helpful but are looking for additional information to kickstart your calculation building needs, either contact our team or check out the helpful resources listed below:

Support Documentation

- <u>Calculations Overview</u>
- Writing a Calculation
- Using the Calculation Builder

Helpful Blogs

- Feature Spotlight: The Calculation Builder
- <u>3 Calculations to Add to Your Forms Today</u>
- <u>ProTip: How to Use Conditional Logic in Automatic Calculations</u>
- ProTip: Use Barcode Scanning to Populate Your Mobile Form Fields
- <u>ProTip: Use Database Fields to Auto-Fill Your Forms</u>
- ProTip: Show or Hide Fields Based on Form Ownership

How GoFormz users are using Calculations

- <u>Case Study: Oscar W. Larson</u>
- <u>Case Study: Muller Management</u>
- <u>Case Study: Energy Efficient Resources</u>
- <u>Case Study: Kaw Valley Greenhouses</u>
- Case Study: Lennon, Smith, Souleret Engineering, Inc.
- <u>Case Study: Romo Durable Graphics</u>



To learn more, visit www.goformz.com

proservices@goformz.com

sales@goformz.com