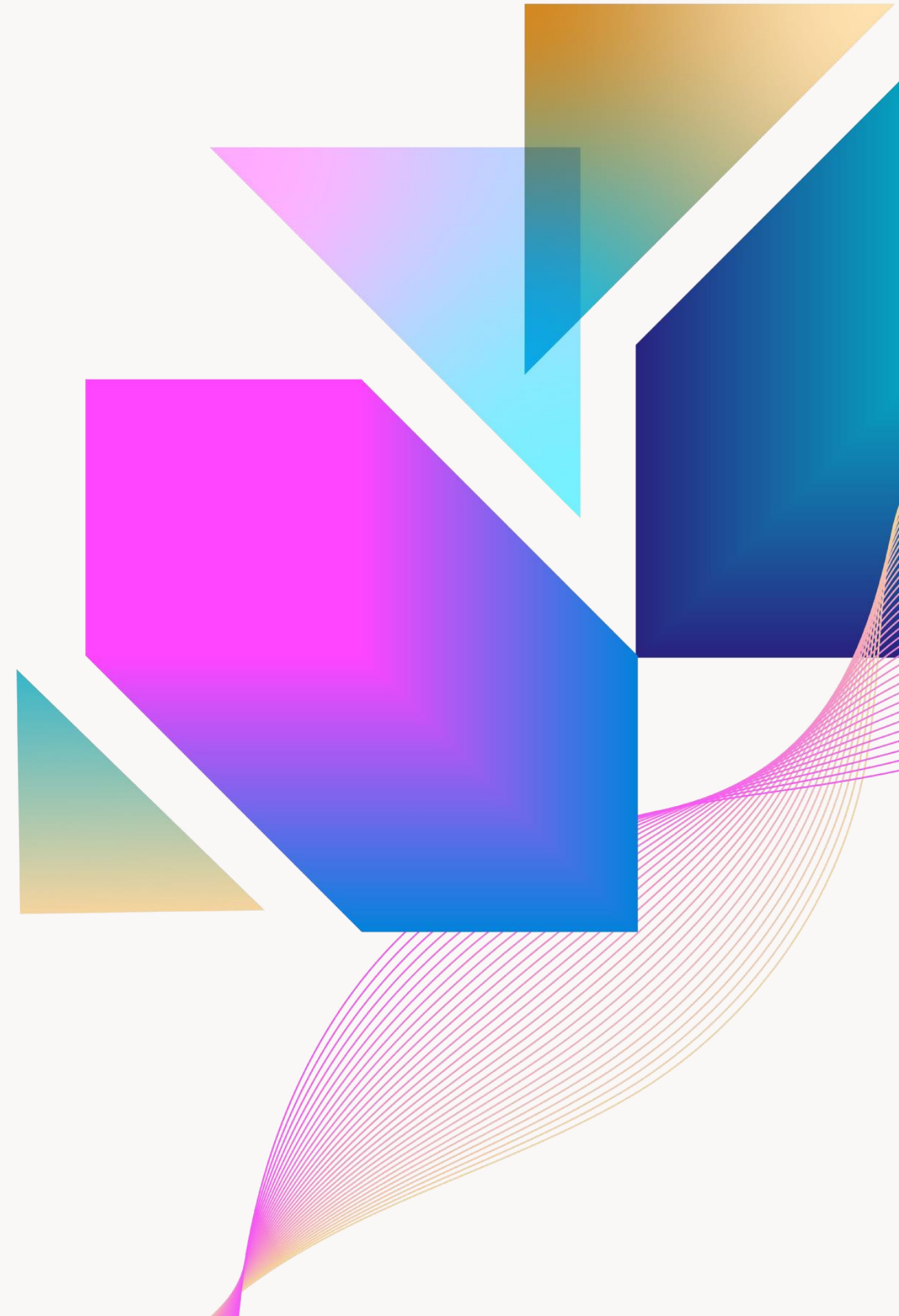




# Mastering Dynamic Stamps in Bluebeam Revu

Customization with JavaScript

Elizabeth Larsen, PE



# Speaker Intro

Liz is a structural engineer originally from Rochester, NY, now based in Raleigh, NC. She has nearly 15 years of industry experience and has been using Bluebeam Revu for over a decade.

This is her third time presenting at the Bluebeam conference. She enjoys sharing practical Bluebeam Revu tips, exploring JavaScript applications in Revu, and finding creative ways to use custom statuses in Studio Sessions.



**Liz Larsen, PE**

**Structural Project Engineer**

**Stewart – Raleigh, NC**

# Agenda

01 What are Bluebeam stamps? How do they differ from markups?

02 Types of stamps

- Static vs Dynamic

03 What makes a Dynamic Stamp different?

04 Some examples

- Static
- Basic dynamic
- Modify an existing stamp

05 Tips and Best Practices

# The Basics

An intro to stamps in Bluebeam Revu



# Stamps vs Markups

## Why stamps exist

- **Markups** are editable annotations.
- **Stamps** are locked down and not easily edited.
- Useful for approvals, reviews, and consistency.
- Can be **static** (same every time) or **dynamic** (changes automatically).
- Stamps are just PDFs stored in your **Stamps** folder (Preferences → Tools)

### CONTRACTOR'S SHOP DRAWING REVIEW

Review is for general conformance with contract documents. Sole responsibility for correctness of dimensions, details, quantities, and safety during fabrication and erection shall remain with the Subcontractor. Sub to notify GC if discrepancies arise and/or if coordination is required with other trades.

Reviewed by: \_\_\_\_\_ Date: \_\_\_\_\_

Company Name _____		Company Logo _____
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
APPROVED	REVISE	REJECTED
BY _____	DATE _____	
SUBMITTAL# _____	SPEC _____	

# Basic Stamp Anatomy

And what makes a dynamic stamp different

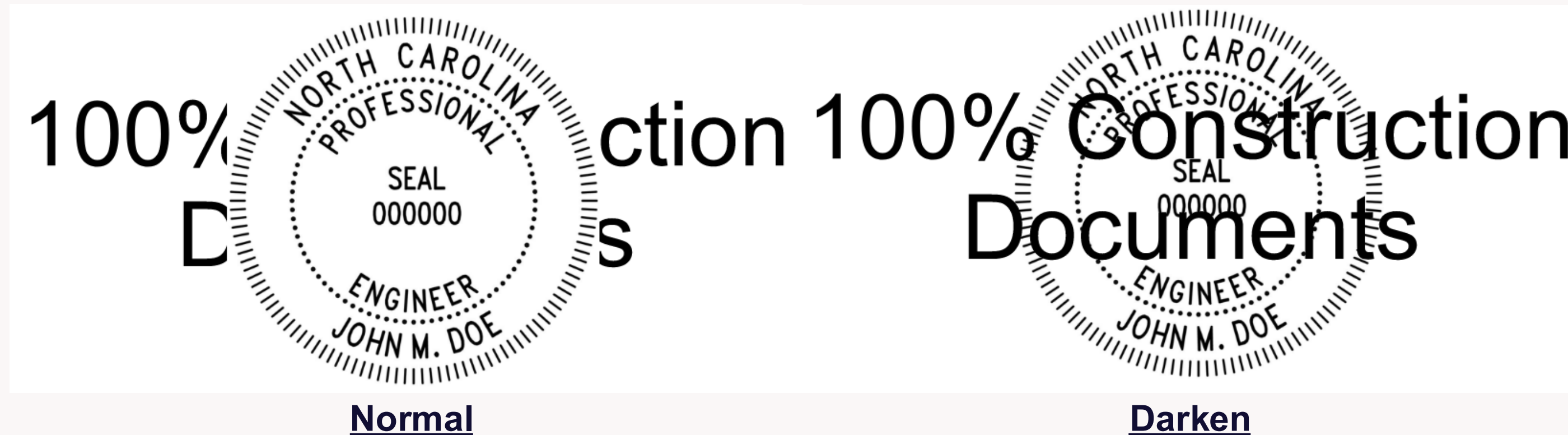
- A **static stamp** is just a PDF
- A **dynamic stamp** has form fields and JavaScript
- Both types are stored in your Bluebeam **Stamps folder**

The image displays two components of the Bluebeam software interface. On the left is a dark-themed stamp library panel. It shows a tree view with 'Page 1' containing a 'Status' group with radio buttons for 'Approved', 'Revise', and 'Rejected'. Below this are several text stamps: 'CheckedBy', 'Date', 'Submittal', 'Spec', 'Project', and 'Calculation'. The 'CheckedBy' stamp is currently selected. On the right is a preview of a dynamic stamp form. This form includes a 'Company Name' label and a text input field, a 'Company Logo' label and a rectangular box, three checkboxes labeled 'APPROVED', 'REVISE', and 'REJECTED' (with 'REJECTED' checked), a 'BY' label followed by a text input field, a 'DATE' label followed by a text input field, a 'SUBMITTAL#' label followed by a text input field, and a 'SPEC' label followed by a text input field. The form is enclosed in a thick black border.

# Basic Stamp Anatomy, con't

## Overlay settings

- You can use “blend modes” to control how they sit on top of drawings
  - Normal – non-transparent background
  - Darken – best for when the document needs to be seen through the stamp



# Simple Example

No coding needed

- Start by placing a normal text markup
- Select a dynamic token from the dropdown like **&date**
- Revu replaces it with today's date when you place the stamp
- No JavaScript needed — it just works

**&[Date]**

Dynamic ▼

User

File

File Without Extension

Path

Date

Time

Day

Month (2 Digit)

Month (3 Letter Abbreviation)

Month (Full Name)

Year (2 Digit)

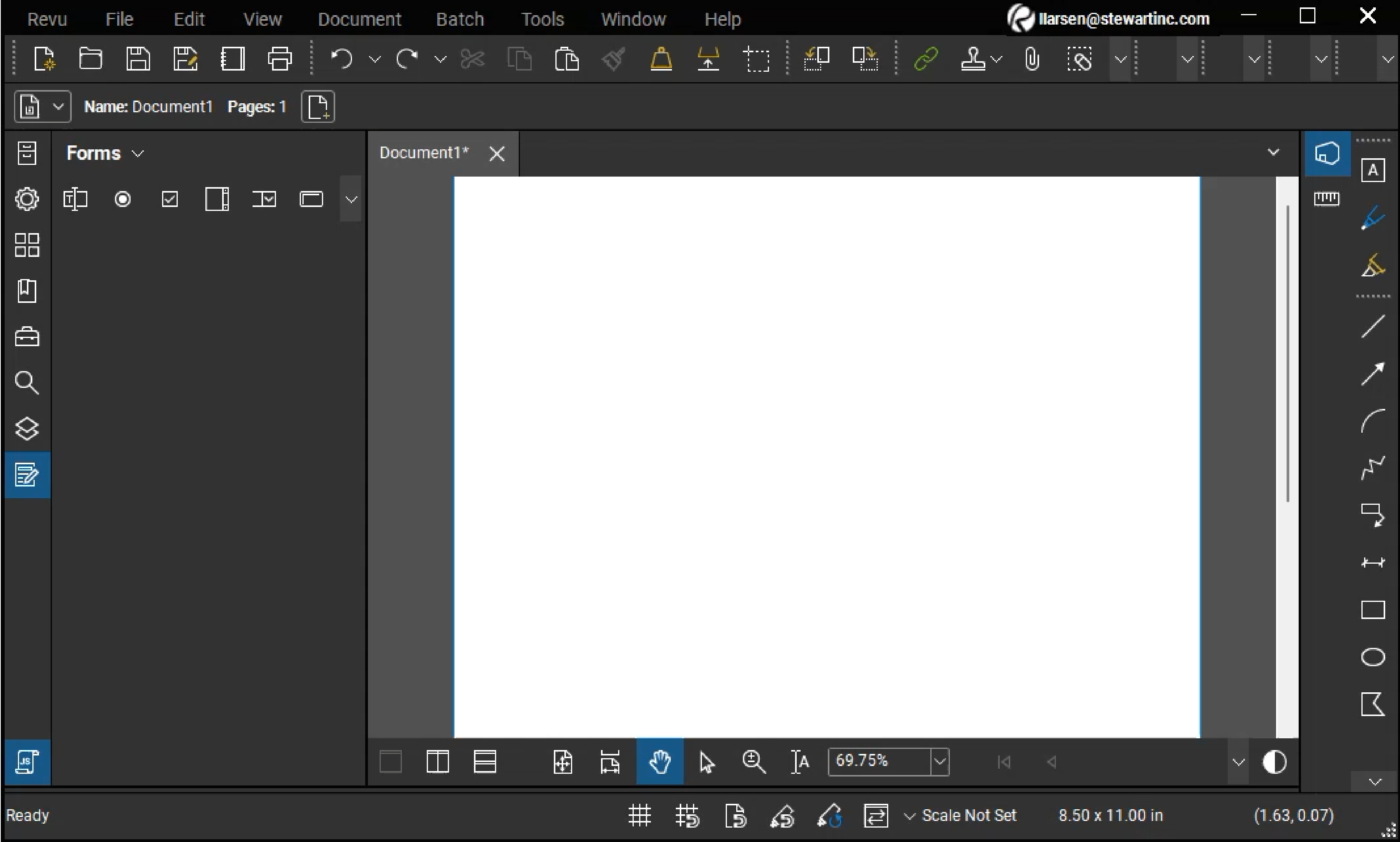
Year (4 Digit)

Hour

minutes

Seconds





# A Simple JavaScript Example

## A basic dynamic stamp

- You can place basic JavaScript right in the form field.
- Always add `event.source` && `event.source.forReal` so the code will only run when the stamp is placed, not when you're editing the stamp.
- The example code below checks if today is a weekend or weekday.

```
if (event.source && event.source.forReal) {  
  var d = new Date().getDay(); // Sunday=0 ... Saturday=6  
  event.value = (d === 0 || d === 6) ? "Weekend" : "Weekday"; }  
}
```



Review is for general conformance with contract documents. Sole responsibility for correctness of dimensions, details, quantities, and safety during fabrication and erection shall remain with the Subcontractor. Sub to notify GC if discrepancies arise and/or if coordination is required with other trades.

Reviewed by: \_\_\_\_\_ Date: 09/04/2025

# Batch vs Non-batch

- **Non-batch:** prompts you *every time* you place the stamp
- **Batch:** remembers your last inputs and reuses them until you close and re-open Revu
- Behavior depends on how the stamp's script is written
- Both are useful in different situations

## Non-batch Stamp

Stamp Dialog

JavaScript Window

Status

☐ Approved ☐ Revise ☐ Rejected ☐ Not Reviewed

Options

Project:

Checked by:

Date:

Submittal #:

Spec #:

OK

1<sup>st</sup> placement

Stamp Dialog

JavaScript Window

Status

☐ Approved ☐ Revise ☐ Rejected ☐ Not Reviewed

Options

Project:

Checked by:

Date:

Submittal #:

Spec #:

OK

2<sup>nd</sup> placement

## Batch Stamp

Stamp Dialog

JavaScript Window

Status

☐ Approved ☐ Revise ☐ Rejected ☐ Not Reviewed

Options

Project:

Checked by:

Date:

Submittal #:

Spec #:

OK

1<sup>st</sup> placement

Stamp Dialog

JavaScript Window

Status

☒ Approved ☐ Revise ☐ Rejected ☐ Not Reviewed

Options

Project:

Checked by:

Date:

Submittal #:

Spec #:

OK

2<sup>nd</sup> placement



# Modifying Dynamic Stamps

Start from an existing stamp and make  
it your own





# Modifying existing stamps

You don't need to reinvent the wheel

- Start with a stamp that already does something close to what you need
- Bluebeam provides sample dynamic stamps on their website
- Online communities share free stamps with different features
- Easier to modify than to build from scratch
- Less likely for a typo to ruin your efforts

5. The next time you select the Stamp tool, your imported stamp(s) will appear in the list.



For more information about customizing and editing your interactive stamp(s), please visit [this article](#).

## Assigned Discipline

Discipline Review	Company Name/Logo
Reviewing Discipline:	
Reviewed By:	Date:

## Engineering Review

COMPANY NAME	COMPANY LOGO
<input type="checkbox"/> APPROVED	
<input type="checkbox"/> APPROVED AS CORRECTED	
<input type="checkbox"/> REVISE AND RESUBMIT	
<input type="checkbox"/> REJECTED	
<input type="checkbox"/> FOR INFORMATION ONLY	
DISCLAIMER	
BY	DATE

## Submittal Project

COMPANY LOGO	Company Name Project Name	DISCLAIMER
By:	Architect:	
Date:	Submittal No:	
Supplier:	Project No:	

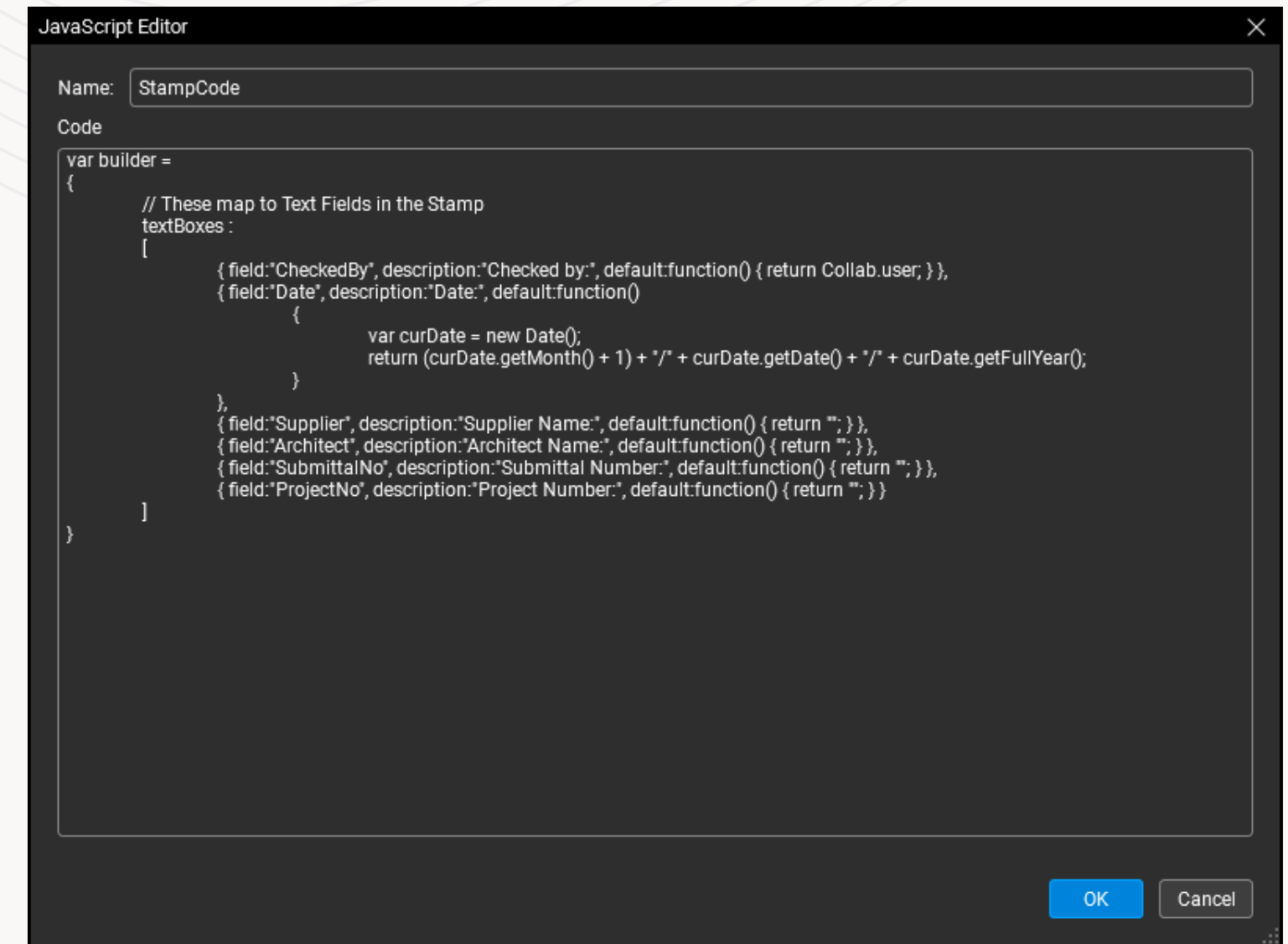
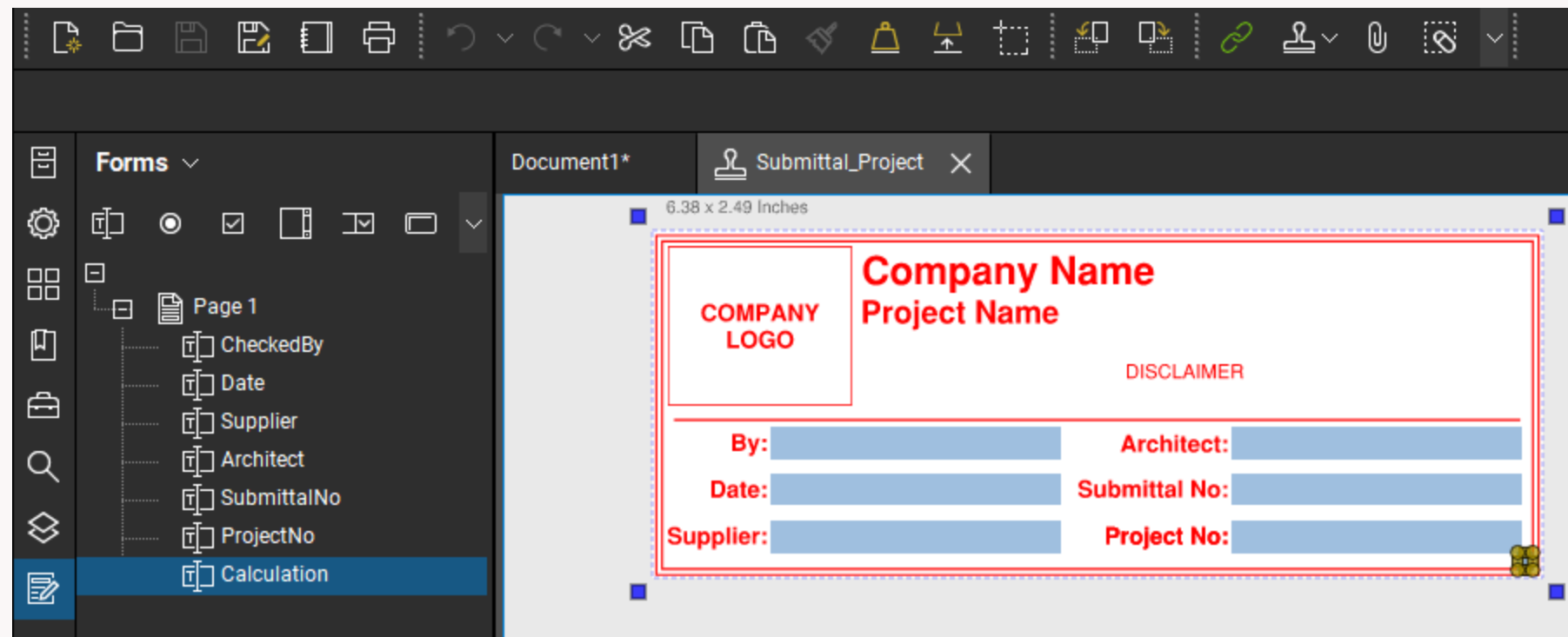
## Submittal Status

Company Name	Company Logo	
<input type="checkbox"/> APPROVED	<input type="checkbox"/> REVISE	<input type="checkbox"/> REJECTED
BY	DATE	
SUBMITTAL#	SPEC	

# Modifying existing stamps, con't

## Stamp code vs calculation code

- **Stamp code (CTRL+SHIFT+J)** = Define what Bluebeam should ask the user for (the “builder” object in the code)
- **Calculation code** = the engine that runs the dialog box
- Rule of thumb: edit stamp code, leave calc code alone



Stamp code

Calc code

# Modifying existing stamps, con't

How to update the stamp and the code

- Open stamp PDF in Revu
- Add or rename form fields: **names must match builder prompts exactly**
- Tell Revu what to prompt for by updating the builder object (ie the stamp code)
- Test the stamp on a blank page before putting it into production.

**Let's walk  
through an  
example**



# The stamp

The stamp

Project Name

New Federal Building

Company Logo

☒ APPROVED

☐ REJECTED

☐ REVISE

☐ NOT REVIEWED

BY Liz

DATE 2/20/2023

SUBMITTAL# 1111

SPEC 02 25 00

This review is only for general conformance of the project and general compliance. Corrections or comments made on these drawings during this review do not relieve Contractor/Subcontractor from compliance with the requirements of the plans and specifications. Contractor is responsible for all dimensions and fabrication to be confirmed and correlated at the job site

The dialog box

Stamp Dialog

JavaScript Window

Status

☒ Approved ☐ Revise ☐ Rejected ☐ Not Reviewed

Options

Project:

New Federal Building

Checked by:

Liz

Date:

2/20/2023

Submittal #:

1111

Spec #:

02 25 00

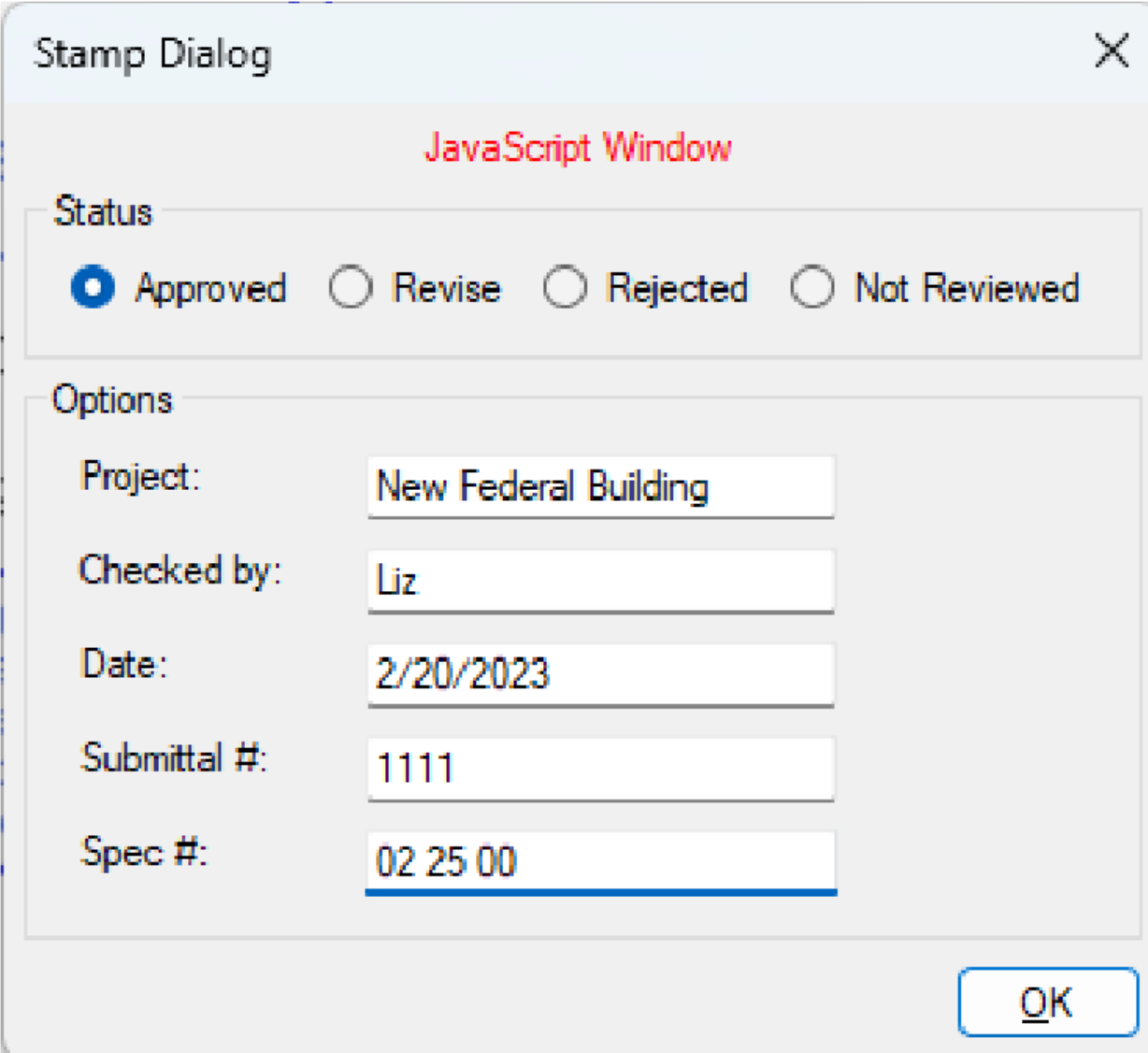
OK



# The code (CTRL+SHIFT+J)

It looks overwhelming, but let's break it down

```
1  var builder = {
2    // This maps to a Radio Group in the PDF named 'Status'
3    radioGroup: "Status",
4    radioButtons: [
5      // value maps to the 'Choice' of each radio button in the group, description will show on the dialog
6      { value: "Approved", description: "Approved" },
7      { value: "Revise", description: "Revise" },
8      { value: "Rejected", description: "Rejected" },
9      { value: "NotReviewed", description: "Not Reviewed" }
10   ],
11   radioErrorMsg: "Please select a status",
12
13   // These map to Text Fields in the Stamp
14   textBoxes: [
15     { field: "Project", description: "Project:", default: function() { return ""; } },
16     { field: "CheckedBy", description: "Checked by:", default: function() { return Collab.user; } },
17     { field: "Date",
18       description: "Date:",
19       default: function() {
20         var curDate = new Date();
21         return (curDate.getMonth() + 1) + "/" + curDate.getDate() + "/" + curDate.getFullYear();
22       } },
23     { field: "Submittal", description: "Submittal #:", default: function() { return ""; } },
24     { field: "Spec", description: "Spec #:", default: function() { return ""; } }
25   ],
26 }
```



The screenshot shows a 'Stamp Dialog' window with a title bar and a close button. Inside, there's a 'JavaScript Window' section. The 'Status' section contains four radio buttons: 'Approved' (selected), 'Revise', 'Rejected', and 'Not Reviewed'. The 'Options' section contains five text input fields: 'Project' (filled with 'New Federal Building'), 'Checked by' (filled with 'Liz'), 'Date' (filled with '2/20/2023'), 'Submittal #' (filled with '1111'), and 'Spec #' (filled with '02 25 00'). An 'OK' button is at the bottom right.

# The code

It looks overwhelming, but let's break it down

```
1  var builder = {
2    // This maps to a Radio Group in the PDF named 'Status' ←
3    radioGroup: "Status",
4    radioButtons: [
5      // value maps to the 'Choice' of each radio button in the group, description will show on the dialog
6      { value: "Approved", description: "Approved" },
7      { value: "Revise", description: "Revise" },
8      { value: "Rejected", description: "Rejected" },
9      { value: "NotReviewed", description: "Not Reviewed" }
10   ],
11   radioErrorMsg: "Please select a status",
12
13   // These map to Text Fields in the Stamp
14   textBoxes: [
15     { field: "Project", description: "Project:", default: function() { return ""; } },
16     { field: "CheckedBy", description: "Checked by:", default: function() { return Collab.user; } },
17     { field: "Date",
18       description: "Date:",
19       default: function() {
20         var curDate = new Date();
21         return (curDate.getMonth() + 1) + "/" + curDate.getDate() + "/" + curDate.getFullYear();
22       } },
23     { field: "Submittal", description: "Submittal #:", default: function() { return ""; } },
24     { field: "Spec", description: "Spec #:", default: function() { return ""; } }
25   ],
26 }
```

The screenshot shows a 'Stamp Dialog' window with a title bar and a close button. Inside, there's a 'JavaScript Window' section. The 'Status' section contains four radio buttons: 'Approved' (selected), 'Revise', 'Rejected', and 'Not Reviewed'. The 'Options' section contains five text input fields: 'Project' (New Federal Building), 'Checked by' (Liz), 'Date' (2/20/2023), 'Submittal #' (1111), and 'Spec #' (02 25 00). An 'OK' button is at the bottom right.

# The code

It looks overwhelming, but let's break it down

```
1  var builder = {
2    // This maps to a Radio Group in the PDF named 'Status' ←
3    radioGroup: "Status",
4    radioButtons: [
5      // value maps to the 'Choice' of each radio button in the group, description will show on the dialog
6      { value: "Approved", description: "Approved" },
7      { value: "Revise", description: "Revise" },
8      { value: "Rejected", description: "Rejected" },
9      { value: "NotReviewed", description: "Not Reviewed" }
10   ],
11   radioErrorMsg: "Please select a status",
12
13   // These map to Text Fields in the Stamp ←
14   textBoxes: [
15     { field: "Project", description: "Project:", default: function() { return ""; } },
16     { field: "CheckedBy", description: "Checked by:", default: function() { return Collab.user; } },
17     { field: "Date",
18       description: "Date:",
19       default: function() {
20         var curDate = new Date();
21         return (curDate.getMonth() + 1) + "/" + curDate.getDate() + "/" + curDate.getFullYear();
22       } },
23     { field: "Submittal", description: "Submittal #:", default: function() { return ""; } },
24     { field: "Spec", description: "Spec #:", default: function() { return ""; } }
25   ],
26 }
```

The screenshot shows a 'Stamp Dialog' window with a title bar and a close button. The dialog is divided into two main sections: 'Status' and 'Options'. The 'Status' section contains a radio group with four options: 'Approved' (selected), 'Revise', 'Rejected', and 'Not Reviewed'. The 'Options' section contains five text input fields: 'Project' (containing 'New Federal Building'), 'Checked by' (containing 'Liz'), 'Date' (containing '2/20/2023'), 'Submittal #' (containing '1111'), and 'Spec #' (containing '02 25 00'). An 'OK' button is located at the bottom right of the dialog.



# The code

It looks overwhelming, but let's break it down

```
2 // This maps to a Radio Group in the PDF named 'Status'
3 radioGroup: "Status", ←
4 radioButtons: [
5     // value maps to the 'Choice' of each radio button in the group, description will show on the dialog
6     { value: "Approved", description: "Approved" },
7     { value: "Revise", description: "Revise" },
8     { value: "Rejected", description: "Rejected" },
9     { value: "NotReviewed", description: "Not Reviewed" }
10 ],
11 radioErrorMsg: "Please select a status",
```

radioGroup: "Status" → radio group name

The radioGroup text must match the form field name exactly

value: "Approved" → the form field name

The value text must match the form field name exactly

description: "Approved" → the label in the dialog box

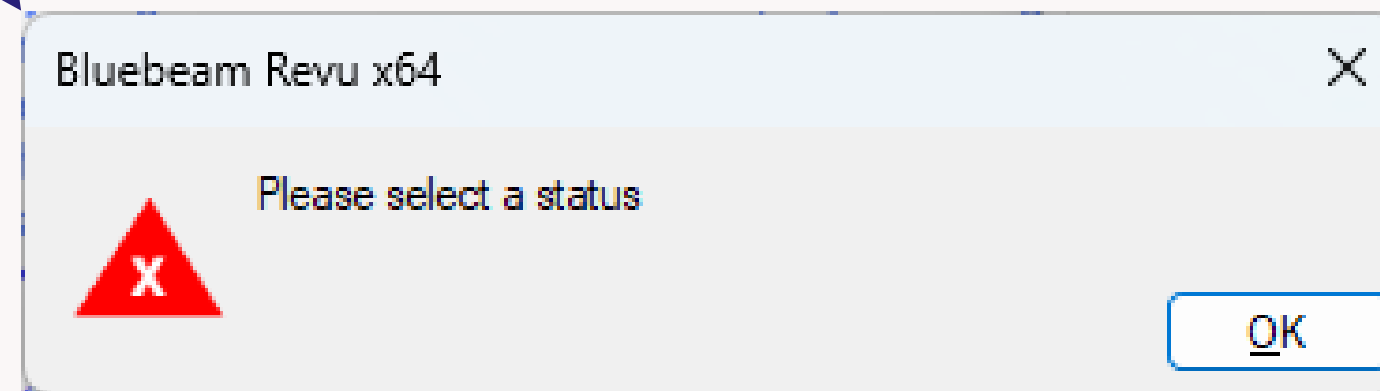
The screenshot shows a 'Stamp Dialog' window titled 'JavaScript Window'. It contains a 'Status' section with four radio buttons: 'Approved' (selected), 'Revise', 'Rejected', and 'Not Reviewed'. Below this is an 'Options' section with several text input fields: 'Project' (New Federal Building), 'Checked by' (Liz), 'Date' (2/20/2023), 'Submittal #' (1111), and 'Spec #' (02 25 00). An 'OK' button is at the bottom right.

The screenshot shows a 'Forms' panel with a tree view of form elements. The 'Status' element is highlighted with a purple arrow. Below it, the radio buttons for 'Approved', 'Revise', 'Rejected', and 'NotReviewed' are listed. Other elements in the tree include 'CheckedBy', 'Date', 'Submittal', 'Spec', 'Project', and 'Calculation'.

# The code

It looks overwhelming, but let's break it down

```
2 // This maps to a Radio Group in the PDF named 'Status'
3 radioGroup: "Status",
4 radioButtons: [
5     // value maps to the 'Choice' of each radio button in the group, description will show on the dialog
6     { value: "Approved", description: "Approved" },
7     { value: "Revise", description: "Revise" },
8     { value: "Rejected", description: "Rejected" },
9     { value: "NotReviewed", description: "Not Reviewed" }
10 ],
11 radioErrorMsg: "Please select a status",
```





# The code

It looks overwhelming, but let's break it down

```
1  var builder = {
2    // This maps to a Radio Group in the PDF named 'Status'
3    radioGroup: "Status",
4    radioButtons: [
5      // value maps to the 'Choice' of each radio button in the group, description will show on the dialog
6      { value: "Approved", description: "Approved" },
7      { value: "Revise", description: "Revise" },
8      { value: "Rejected", description: "Rejected" },
9      { value: "NotReviewed", description: "Not Reviewed" }
10   ],
11   radioErrorMsg: "Please select a status",
12
13   // These map to Text Fields in the Stamp
14   textBoxes: [
15     { field: "Project", description: "Project:", default: function() { return ""; } },
16     { field: "CheckedBy", description: "Checked by:", default: function() { return Collab.user; } },
17     { field: "Date",
18       description: "Date:",
19       default: function() {
20         var curDate = new Date();
21         return (curDate.getMonth() + 1) + "/" + curDate.getDate() + "/" + curDate.getFullYear();
22       } },
23     { field: "Submittal", description: "Submittal #:", default: function() { return ""; } },
24     { field: "Spec", description: "Spec #:", default: function() { return ""; } }
25   ]
26 }
```

The screenshot shows a 'Stamp Dialog' window with a title bar and a close button. Inside, there's a 'JavaScript Window' section. The 'Status' section contains four radio buttons: 'Approved' (selected), 'Revise', 'Rejected', and 'Not Reviewed'. The 'Options' section contains five text input fields: 'Project' (filled with 'New Federal Building'), 'Checked by' (filled with 'Liz'), 'Date' (filled with '2/20/2023'), 'Submittal #' (filled with '1111'), and 'Spec #' (filled with '02 25 00'). An 'OK' button is at the bottom right.

# The code

It looks overwhelming, but let's break it down

```
13 // These map to Text Fields in the Stamp ←
14 textBoxes: [
15   { field: "Project", description: "Project:", default: function() { return ""; } },
16   { field: "CheckedBy", description: "Checked by:", default: function() { return Collab.user; } },
17   { field: "Date",
18     description: "Date:",
19     default: function() {
20       var curDate = new Date();
21       return (curDate.getMonth() + 1) + "/" + curDate.getDate() + "/" + curDate.getFullYear();
22     } },
23   { field: "Submittal", description: "Submittal #:", default: function() { return ""; } },
24   { field: "Spec", description: "Spec #:", default: function() { return ""; } }
25 ]
```

field: "Project" → text form field name

- The field text must match the form field name exactly
- This is true for all text fields

description: "Project" → the label in the dialog box

Optional

default: function() { return "" ; } → put text in the quotes to set a default value, can be overwritten

The screenshot shows the Bluebeam Forms tool interface. At the top, there's a 'Forms' dropdown menu. Below it, a toolbar contains icons for various form elements: a text field, a radio button, a checkbox, a list box, and a dropdown menu. The main area displays a form layout for a stamp. It starts with a 'Page 1' header. Below that is a 'Status' section with four radio buttons labeled 'Approved', 'Revise', 'Rejected', and 'NotReviewed'. Underneath the status section is a list of text fields: 'CheckedBy', 'Date', 'Submittal', 'Spec', 'Project', and 'Calculation'. A purple arrow points from the 'Project' field in the code block to the 'Project' text field in the form layout.

# The code

It looks overwhelming, but let's break it down

```
13 // These map to Text Fields in the Stamp ←
14 textBoxes: [
15   { field: "Project", description: "Project:", default: function() { return ""; } },
16   { field: "CheckedBy", description: "Checked by:", default: function() { return Collab.user; } },
17   { field: "Date",
18     description: "Date:",
19     default: function() {
20       var curDate = new Date();
21       return (curDate.getMonth() + 1) + "/" + curDate.getDate() + "/" + curDate.getFullYear(); },
22   { field: "Submittal", description: "Submittal #:", default: function() { return ""; } },
23   { field: "Spec", description: "Spec #:", default: function() { return ""; } }
24 ],
```

default: function() { return Collab.user ; }

→ This auto fills the user's *Windows username* in the dialog box

default: function() { return identity.name ; }

→ This auto fills the user's *Bluebeam username* in the dialog box

The screenshot shows the 'Forms' tool palette in Bluebeam. At the top, there are icons for text fields, radio buttons, checkboxes, and other form controls. Below these, a sample form layout is displayed. The form includes a 'Page 1' header, a 'Status' section with four radio button options: 'Approved', 'Revise', 'Rejected', and 'NotReviewed'. Below the status section, there are six text field labels: 'CheckedBy', 'Date', 'Submittal', 'Spec', 'Project', and 'Calculation'. A purple arrow points from the 'Submittal' label in the code block to the 'Submittal' text field in the form layout.

# The code

It looks overwhelming, but let's break it down

```
13 // These map to Text Fields in the Stamp ←
14 textBoxes: [
15   { field: "Project", description: "Project:", default: function() { return ""; } },
16   { field: "CheckedBy", description: "Checked by:", default: function() { return Collab.user; } },
17   { field: "Date",
18     description: "Date:",
19     default: function() {
20       var curDate = new Date();
21       return (curDate.getMonth() + 1) + "/" + curDate.getDate() + "/" + curDate.getFullYear(); },
22   { field: "Submittal", description: "Submittal #:", default: function() { return ""; } },
23   { field: "Spec", description: "Spec #:", default: function() { return ""; } }
24 ],
```

This function auto fills today's date in the dialog box, but it can be overwritten when the user places the stamp.

The code above uses the MM/DD/YYYY format.

You can rearrange the three curDate portions to format the date however you need:

- (curDate.getMont()+1) → current month
- curDate.getDate() → current day of the month
- curDate.getFullYear() → current year

The screenshot shows the Bluebeam Forms tool interface. At the top, there's a 'Forms' dropdown menu. Below it, a toolbar contains various form field icons: a text box, a radio button, a checkbox, a list box, and a dropdown menu. The main area displays a form layout for a stamp. It starts with a 'Page 1' header, followed by a 'Status' section with four radio button options: 'Approved', 'Revise', 'Rejected', and 'NotReviewed'. Below the status section are five text input fields, each with a text box icon to its left. These fields are labeled 'CheckedBy', 'Date', 'Submittal', 'Spec', and 'Calculation'. A purple arrow points from the 'Date' field in the code block to the 'Date' text input field in the form layout.



# Some notes about code formatting



# The code - again

Let's talk about formatting (aka syntax)

```
1  var builder = {
2    // This maps to a Radio Group in the PDF named 'Status'
3    radioGroup: "Status",
4    radioButtons: [
5      // value maps to the 'Choice' of each radio button in the group, description will show on the dialog
6      { value: "Approved", description: "Approved" },
7      { value: "Revise", description: "Revise" },
8      { value: "Rejected", description: "Rejected" },
9      { value: "NotReviewed", description: "Not Reviewed" }
10   ],
11   radioErrorMsg: "Please select a status",
12
13   // These map to Text Fields in the Stamp
14   textBoxes: [
15     { field: "Project", description: "Project:", default: function() { return ""; } },
16     { field: "CheckedBy", description: "Checked by:", default: function() { return Collab.user; } },
17     { field: "Date",
18       description: "Date:",
19       default: function() {
20         var curDate = new Date();
21         return (curDate.getMonth() + 1) + "/" + curDate.getDate() + "/" + curDate.getFullYear();
22       },
23     { field: "Submittal", description: "Submittal #:", default: function() { return ""; } },
24     { field: "Spec", description: "Spec #:", default: function() { return ""; } }
25   ],
26 }
```

- Brackets and parentheses always come in pairs
  - { } = groups of settings
  - [ ] = lists of items
  - ( ) = function calls
- Commas separate items in the same group
- Missing or extra characters will break the stamp
- [jshint.com](https://jshint.com) is an excellent resource to check for any formatting mistakes

# Live demo time!

Because live demos always go perfectly

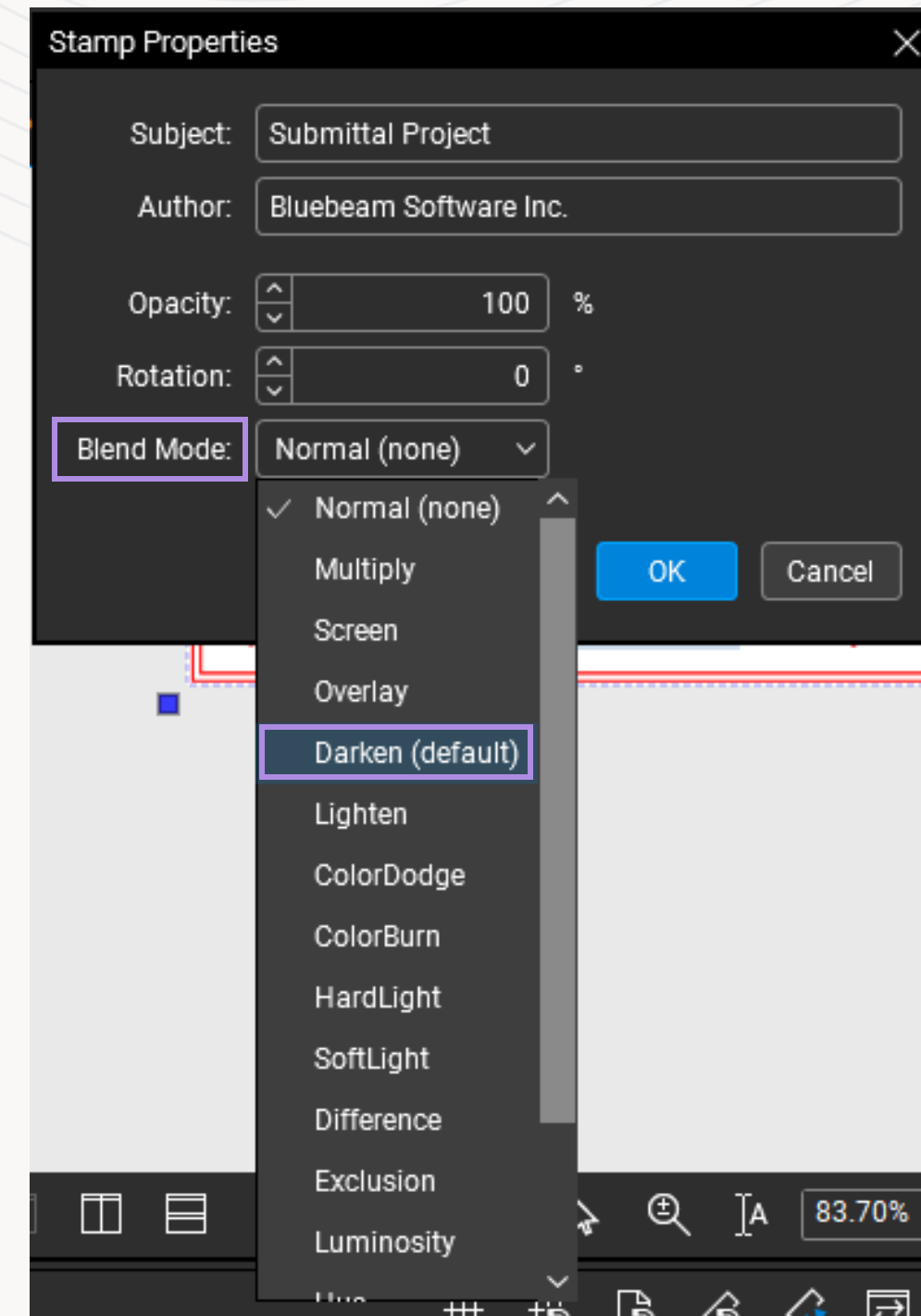
# Tips and Best Practices

Set yourself up for success

# Tips & Best Practices

## Set yourself up for success

- Always copy first – never edit the only copy
- Start from a Bluebeam template stamp or an existing stamp that has the features you want.
  - These stamps come with a calc code
  - Without a calc code, the stamp will not work
- Test in a dummy document for:
  - size
  - transparency → update Blend Mode to Darken
  - user input functionality





# Resources

## Where to go from here

- Bluebeam dynamic stamp instructions:
  - <https://support.bluebeam.com/revu/how-to/add-and-edit-interactive-stamps.html>
- Download Bluebeam example stamps:
  - <https://support.bluebeam.com/revu/resources/revu-custom-libraries.html#Stamps>
- ATG video about dynamic stamps:
  - [https://www.youtube.com/watch?v=NxWDot\\_kQfY](https://www.youtube.com/watch?v=NxWDot_kQfY)

# Q&A

# Thank you

Liz Larsen, PE

Structural Project Engineer

Stewart | Raleigh, NC

[www.lizlarsen.me](http://www.lizlarsen.me)

[www.linkedin.com/in/liz-larsen](http://www.linkedin.com/in/liz-larsen)



[www.lizlarsen.me](http://www.lizlarsen.me)



[www.linkedin.com/in/liz-larsen](http://www.linkedin.com/in/liz-larsen)