# Select Button Checkmark and Radio Extension

Last Modified on 07/30/2026 5:57 pm EDT

The **Select Button Checkmark and Radio Extension** allows you to add styled checkmarks or radio-style indicators to Digioh select buttons, including quiz answer buttons and generic select buttons.

This extension is fully configurable using metadata and does not require custom CSS, images, or HTML modifications.

**What You'll Learn in This Article:**

- How to install and enable the Select Button Checkmark and Radio Extension
- How to configure checkmark and radio styles using metadata
- How to test and troubleshoot the extension

## Overview

This extension enhances Digioh buttons by dynamically rendering visual selection indicators on select button elements, based on metadata configuration. It's important to note that these visuals will not display in the editor, only in preview or live on your site.  Follow the instructions below to set up your buttons, then Preview to test.

## Use Cases

- Quiz answer buttons
- Generic Digioh select buttons
- Survey style button selections

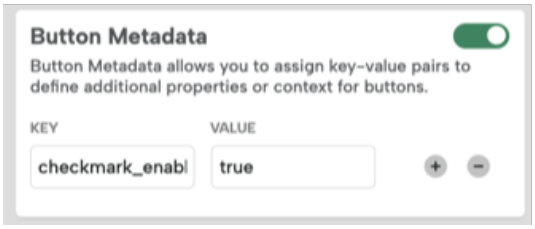
## How to Install

1. Log into your Digioh account.
2. Open the **Extensions** section from the profile menu (top-right dropdown).
3. Search for **"Select Button Checkmark and Radio"** in the available extensions list (under Button Customization).
4. Click **Install**.
5. **Publish** the Extensions Page to your Site

## Enable a Checkmark on a Button

Add the following metadata at the **button (element) level**:

**checkmark_enable = true**

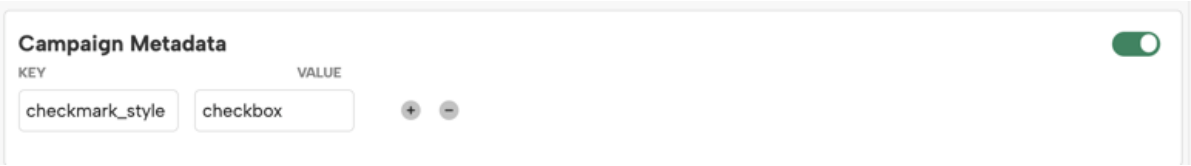



This must be applied individually to each button that should display a checkmark.

## Checkmark Styles (Campaign-Level Metadata)

Set the style at the **campaign level metadata**

checkmark_style = [style name]




### Available Styles

**checkbox** Displays a visible square box at all times. The checkmark appears inside the box when selected.

**textmark** (default) Displays a checkmark inline next to the button text when selected.

**indent** Reserves space for the checkmark even before selection to prevent layout shift.

**shift** Does not reserve space. The checkmark appears when selected and shifts the text.

If no style is set, the extension defaults to: **textmark**

## Optional Checkmark Customization (Campaign Level Metadata)

You may further customize the checkmark appearance using the following metadata.

### checkmark_icon

checkmark_icon = ✓

- Accepts a character (e.g., ✓)
- Or a direct image URL (https://example.com/icon.png)

Default: ✓

### checkmark_position

checkmark_position = left

Options:

- left
- right

Default: left

### checkmark_color

checkmark_color = #000

- Any valid hex color
- Default behavior:
  - #000 for checkbox style
  - inherits button text color for all other styles

### checkmark_size

checkmark_size = 16px

- Accepts common CSS size units (px, em, rem, %)
- Default: Matches button font size

## Checkbox Style Options

(Only applies when checkmark_style = checkbox)

### checkbox_border_color

checkbox_border_color = #000

Default: #000

### checkbox_background

checkbox_background = #ffffff

Default: #fff

### checkbox_size

checkbox_size = 18px

Default: 18px

## Enable Radio Button Style

To render a radio-style selection indicator, add the following metadata at the **button (element)**

**level**:

radio_enable = true

## Radio Customization (Campaign-Level Metadata)

### radio_border_color

radio_border_color = #000

Default: inherits button text color

### radio_fill_color

radio_fill_color = #000

Default: inherits button text color

### radio_size

radio_size = 18px

Default: 18px

## Important Rules and Limitations

- You must apply checkmark_enable or radio_enable individually to each button.
- A button **cannot** use both checkmark_enable and radio_enable. If both are applied, the extension will log an error and skip that button.
- The extension respects the button's existing text alignment.
- Recommended styling: **Left aligned text + checkbox style** for a traditional UI appearance.

## QA and Testing

To test, ensure the extension is added to your account and your campaign buttons are setup. You can Save and Preview your box to see the checkmarks or radio buttons in action.

To test live on your site in Digioh QA Mode:

1. Click **Publish QA** in the Digioh account.
2. Add ?boxqamode to the page URL. (Use &boxqamode if the URL already contains query parameters.)
3. Verify:
    - Indicators render correctly

    - Styles match metadata configuration
    - Indicators persist across page transitions (for multi-page quizzes

4. Confirm the extension is published — unpublished extensions will not run on the test page.

## Troubleshooting

If indicators are not appearing:

- Confirm metadata is applied at the correct level (Campaign vs Element).
- Confirm the extension has been published after any changes.

Ensure buttons include: checkmark_enable='true' or radio_enable='true'

- Check the browser console for Digioh App notification errors.
- Confirm no other Custom JS is overwriting button HTML.
- Ensure checkmark_style is one of:
    - checkbox
    - textmark
    - indent
    - shift