

# Setting Up a Button

## Overview

iReporter supports up to 16 buttons. Each button is independently configured with its own mode, label, messages, webhook target, crash/incident fallbacks, and voice text. The number of visible buttons is controlled by the **Number of Buttons** setting.

## Button Mode

Each button has a **Button Mode** dropdown at the top of its section. The mode controls which fields are shown and how the button behaves when pressed:

Mode	What it does
<b>Standard Button</b>	Sends the three message lines via iRacing chat and Discord webhook. No crash or incident fallback panels.
<b>Crash Detection</b>	Includes a Crash Fallback field. When pressed with an active crash, sends the main messages using crash variables. When no crash is active, sends the fallback message instead. Only available when Crash Capture is enabled in iRacing Options.
<b>Incident Detection</b>	Same as Crash Detection but for incidents. Uses incident variables and the Incident Fallback field. Only available when Incident Capture is enabled.
<b>Blue Flag Helper</b>	A Standard button whose template dropdown shows only Blue Flag Helper templates. Useful for grouping blue flag acknowledgements, warnings, and reports on dedicated buttons without mixing them with other Standard templates.
<b>Voice Only</b>	Fires the voice message only – no text is typed into the iRacing chat box. Useful for quick radio calls that don't need a chat record. The voice input box is always shown regardless of the global Voice Options setting.

Changing the mode also filters the **Message Templates** dropdown to show only templates in the matching category.

# Button Fields

## Label

A descriptive name for the button shown in the settings panel (e.g. Contact Report, Yellow Flag). This is for your reference only and is not included in any sent message.

## Message Templates

The **Message Templates** dropdown appears above the message fields. It shows only templates whose category matches the button's current mode " so a Crash Detection button only shows Crash Detection templates. Select a template to instantly fill in all message fields and voice text.

## Message 1, 2, 3

Up to three message lines can be configured per button. When the button is pressed:

- All three messages are sent in sequence if populated.
- Messages support [Message Variables](#) " placeholders filled with live iRacing data at press time.
- Empty lines (containing only ~) are skipped.

## Webhook

Selects which of the three configured webhooks receives this button's messages. Set to **None** to disable webhook posting for this button.

## Disable Text Logging

When ticked, this button's presses are not written to the log file even if logging is globally enabled.

## Voice (Speak Text)

Appears below the message rows. When Voice Options is globally enabled (or when the button is in **Voice Only** mode), a multi-line **Speak Text** box is shown. Enter the text to be spoken aloud when this button is pressed. Message variables such as `{CAR#}` and `{CRASHCAR}` are supported.

Tick **Disable Voice for this button** to suppress voice output while keeping the speak text saved for future use.

# Crash Fallback

Visible on buttons in **Crash Detection** mode (requires Crash Capture to be enabled in iRacing Options). The green-bordered panel is labelled "*Sent when the button is pressed but no Crash has been detected*" – this message is sent when no crash is currently active. Leave blank to send nothing if no crash is active.

# Incident Fallback

Visible on buttons in **Incident Detection** mode (requires Incident Capture to be enabled). Works identically to the Crash Fallback but activates when no incident is currently active.

# Test Fire

The **TEST FIRE** button at the top of each button section simulates a button press without requiring a physical button press. Useful for testing your webhook delivery and voice output during setup. Test Fire does not type into the iRacing chat box – it only fires voice and webhooks.

# Message Templates (Saved)

The **Message Templates** dropdown (above the message rows) allows you to load a saved template into the current button. Templates are filtered by category to match the button's mode.

Click **Manage Templates** (in the top row of the settings panel) to open the template editor. In the editor you can:

- Set the template **Name** and **Category** (Standard, Crash Detection, Incident Detection, or Voice Only)
- Edit **Msg 1, 2, 3** – the three message lines
- Edit **Failover** – the crash or incident fallback message (shown in orange)
- Edit **Voice** – the speak text (shown in blue, wraps to multiple lines)
- Tick **Voice Disabled** – saves the voice-disabled state with the template
- Use { in any field to open the variable picker
- Reorder templates with the **↑** and **↓** buttons
- Add a blank template with **+ Add New Template**
- Delete a template with **Delete**

To save the current button's configuration as a new template, click **Save as Template** below the voice box. The template's category is automatically set to match the button's current mode.

# Default Templates

iReporter ships with 17 ready-to-use templates covering the most common race reporting scenarios. They are created automatically in `Documents\iReporter\templates.txt` the first time SimHub starts. Each template is pre-assigned to the correct button mode category so it appears in the right dropdown automatically.

## Crash Detection templates

These templates require **Crash Capture** to be enabled in iRacing Options and work best on buttons set to **Crash Detection** mode.

Template	Purpose
<b>Avoidable Contact</b>	Reports avoidable contact to Race Control. Sends three chat lines identifying your car, the car involved ( <code>{CRASHCAR}</code> ), the lap, and the relative direction of contact ( <code>{CRASHDIR}</code> ). The fallback message is sent if no crash is currently active. Includes a spoken voice report with pauses for clarity.

## Incident Detection templates

These templates require **Incident Capture** to be enabled in iRacing Options and work best on buttons set to **Incident Detection** mode.

Template	Purpose
<b>Incident Report Close1</b>	Reports a close incident to Race Control. Identifies your car, the incident car ( <code>{INCCAR}</code> ), and the lap number ( <code>{INCLAP}</code> ). The fallback sends a general incident report if no incident is currently active. Includes a spoken voice report.

## Voice Only templates

These templates work on buttons set to **Voice Only** mode – no text is sent to the iRacing chat box.

Template	Purpose
----------	---------

<b>Call Race Control (Voice)</b>	Speaks a radio-style call to Race Control three times followed by your car number. Use this to open a voice communication when you need Race Control's attention without cluttering the chat. No chat messages are sent.
----------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Standard templates

These templates work on any button set to **Standard Button** mode.

Template	Purpose
<b>Yellow Flag Overtake</b>	Reports being overtaken under a yellow flag. Identifies your car, the lap, and the car ahead ( <code>{CARAHEAD}</code> ) as the likely overtaker. Includes a voice report.
<b>Blue Flag Overtake Issues</b>	Reports that a car is failing to observe the blue flag while you are lapping it. Identifies the car ahead ( <code>{CARAHEAD}</code> ) as the offender. Includes a voice report.
<b>Car Ahead Blinking</b>	Reports that the car directly ahead is brake-testing or blinking, creating a safety hazard. Identifies the car ahead ( <code>{CARAHEAD}</code> ). Includes a voice report with pauses.
<b>Sorry Car Behind</b>	Sends a private in-car message directly to the car behind ( <code>{CARBEHIND}</code> ) saying "Sorry â€” My fault!". Voice is disabled â€” chat only, no spoken output.
<b>Sorry Car Ahead</b>	Sends a private in-car message directly to the car ahead ( <code>{CARAHEAD}</code> ) saying "Sorry â€” My fault!". Voice is disabled â€” chat only, no spoken output.
<b>Blue Flag Observed</b>	Notifies the car behind ( <code>{CARBEHIND}</code> ) that you have seen the blue flag and will let them past when it is safe to do so. Two chat messages. Voice is disabled.
<b>Pitting This LAP</b>	Announces that your car is pitting this lap. Voice is disabled â€” use this as a quick notification to nearby cars via chat.
<b>Unsafe Rejoin Ahead</b>	Reports an unsafe track rejoin by the car ahead ( <code>{CARAHEAD}</code> ) to Race Control. Includes a spoken voice report.
<b>Unsafe Rejoin Behind</b>	Reports an unsafe track rejoin by the car behind ( <code>{CARBEHIND}</code> ) to Race Control. Includes a spoken voice report.
<b>Impeding Qualifying Behind</b>	Reports that the car behind ( <code>{CARBEHIND}</code> ) is impeding your qualifying lap. Includes a spoken voice report.
<b>Impeding Qualifying Ahead</b>	Reports that the car ahead ( <code>{CARAHEAD}</code> ) is impeding your qualifying lap. Includes a spoken voice report.
<b>Outlap Overtake Ahead</b>	Reports being overtaken during an outlap by the car ahead ( <code>{CARAHEAD}</code> ). Includes a voice report with pauses.

Template	Purpose
<b>Tow Requested</b>	Notifies Race Control that your car requires a tow. Includes a voice report asking when it is safe to tow.
<b>Bad Driver Standards</b>	Reports poor driving standards to Race Control. Identifies your car, lap, and the three closest cars ( <code>{CLOSECAR1}</code> , <code>{CLOSECAR2}</code> , <code>{CLOSECAR3}</code> ) as likely vehicles involved. Includes a voice report.

# Variable Helper “Type { to Browse Variables

Every message text box (including fields in the Manage Templates editor) has a built-in variable picker. Type an opening curly brace `{` and a list of available variables will appear directly below the text box, each with a short description.

## Filtering the list

Keep typing after the `{` to narrow the list. For example, typing `{CRASH}` will immediately filter the list to show only crash-related variables. The filter is not case-sensitive.

## Inserting a variable

- **Arrow keys** “press `↑` and `↓` to move through the list.
- **Tab or Enter** “inserts the highlighted variable at the cursor position and closes the picker.
- **Click** “click any entry to insert it immediately.
- **Escape** “dismisses the picker without inserting anything.

---

Revision #14

Created 2026-04-22 23:21:37 UTC by DEV API

Updated 2026-04-28 07:52:15 UTC by DEV API