

Button Configuration

Configuring buttons, messages, crash and incident fallbacks

- [Setting Up a Button](#)
- [Message Variables Reference](#)
- [LED Reference](#)
- [Using Conditional Statements in Messages](#)

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 |
|---------------------------|--|
| Standard Button | Sends the three message lines via iRacing chat and Discord webhook. No crash or incident fallback panels. |
| Crash Detection | 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. |
| Incident Detection | Same as Crash Detection but for incidents. Uses incident variables and the Incident Fallback field. Only available when Incident Capture is enabled. |
| Blue Flag Helper | 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. |
| Voice Only | 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 |
|--------------------------|---|
| Avoidable Contact | 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 |
|-------------------------------|---|
| Incident Report Close1 | 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 |
|----------------------------------|--|
| Call Race Control (Voice) | 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 |
|-----------------------------------|---|
| Yellow Flag Overtake | 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. |
| Blue Flag Overtake Issues | 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. |
| Car Ahead Blinking | 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. |
| Sorry Car Behind | 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. |
| Sorry Car Ahead | 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. |
| Blue Flag Observed | 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. |
| Pitting This LAP | Announces that your car is pitting this lap. Voice is disabled â€” use this as a quick notification to nearby cars via chat. |
| Unsafe Rejoin Ahead | Reports an unsafe track rejoin by the car ahead (<code>{CARAHEAD}</code>) to Race Control. Includes a spoken voice report. |
| Unsafe Rejoin Behind | Reports an unsafe track rejoin by the car behind (<code>{CARBEHIND}</code>) to Race Control. Includes a spoken voice report. |
| Impeding Qualifying Behind | Reports that the car behind (<code>{CARBEHIND}</code>) is impeding your qualifying lap. Includes a spoken voice report. |
| Impeding Qualifying Ahead | Reports that the car ahead (<code>{CARAHEAD}</code>) is impeding your qualifying lap. Includes a spoken voice report. |
| Outlap Overtake Ahead | Reports being overtaken during an outlap by the car ahead (<code>{CARAHEAD}</code>). Includes a voice report with pauses. |
| Tow Requested | Notifies Race Control that your car requires a tow. Includes a voice report asking when it is safe to tow. |
| Bad Driver Standards | 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.

Message Variables Reference

Overview

Variables are placeholders wrapped in curly braces that are substituted with live iRacing data when a button is pressed. They can be used in any message field and in the Speak Text field.

If a variable is used in a message but has no current value, the entire message line is suppressed (or replaced by the fallback message if one is configured).

Your Car

| Variable | Description |
|-----------|---|
| {CAR#} | Your car number |
| {CARTYPE} | Your car model / type |
| {CARLAP} | Your current lap (with decimal fraction) |
| {DRIVER} | Your driver name |
| {FLAG} | Current flag colour (e.g. Green, Yellow, Red, Black, Checkered) |

Crash Variables

Available only when a crash is currently active (within the crash timeout window).

| Variable | Description |
|-------------|---|
| {CRASHCAR} | Car number of the crashed car |
| {CRASHNAME} | Driver name of the crashed car |
| {CRASHDIR} | Relative direction of the crash (Ahead / Behind / Left / Right) |
| {CRASHLAP} | Lap number at which the crash occurred |

Incident Variables

Available only when an incident is currently active (within the incident timeout window).

| Variable | Description |
|-----------|---|
| {INCCAR} | Car number involved in the incident |
| {INCNAME} | Driver name involved in the incident |
| {INCDIR} | Relative direction of the incident |
| {INCLAP} | Lap number at which the incident occurred |

Cars Around You

| Variable | Description |
|------------------------------|-------------------------------------|
| {CARAHEAD} / {CARAHEAD1} | Car number of the 1st car ahead |
| {CARAHEAD2} | Car number of the 2nd car ahead |
| {CARAHEAD3} | Car number of the 3rd car ahead |
| {NAMEAHEAD} / {NAMEAHEAD1} | Driver name of the 1st car ahead |
| {NAMEAHEAD2} | Driver name of the 2nd car ahead |
| {NAMEAHEAD3} | Driver name of the 3rd car ahead |
| {CARBEHIND} / {CARBEHIND1} | Car number of the 1st car behind |
| {CARBEHIND2} | Car number of the 2nd car behind |
| {CARBEHIND3} | Car number of the 3rd car behind |
| {NAMEBEHIND} / {NAMEBEHIND1} | Driver name of the 1st car behind |
| {NAMEBEHIND2} | Driver name of the 2nd car behind |
| {NAMEBEHIND3} | Driver name of the 3rd car behind |
| {CLOSECAR1} | Closest car number (any direction) |
| {CLOSECAR2} | 2nd closest car number |
| {CLOSECAR3} | 3rd closest car number |
| {CLOSENAME1} | Closest driver name (any direction) |
| {CLOSENAME2} | 2nd closest driver name |
| {CLOSENAME3} | 3rd closest driver name |
| {GAPHEAD} / {GAPHEAD1} | Gap in seconds to 1st car ahead |
| {GAPHEAD2} | Gap in seconds to 2nd car ahead |
| {GAPHEAD3} | Gap in seconds to 3rd car ahead |

| Variable | Description |
|----------------------------|--|
| {GAPBEHIND} / {GAPBEHIND1} | Gap in seconds to 1st car behind |
| {GAPBEHIND2} | Gap in seconds to 2nd car behind |
| {GAPBEHIND3} | Gap in seconds to 3rd car behind |
| {TOPX} | Comma-separated list of top X race leaders by position (e.g. #3, #11, #27) |

Timing Controls

| Variable | Description |
|-----------|---|
| {PAUSE:N} | Insert a pause of N milliseconds between message segments |

Skip Marker — ~

Enter `~` (tilde) alone in a message slot to skip that slot entirely. No keystrokes are sent.

Conditional Statements

iReporter supports a simplified conditional syntax that lets you include or suppress parts of a message based on live race data. The syntax uses parentheses rather than curly braces, keeping it visually distinct from variables.

For full documentation including AND / OR compound conditions, examples, and the current supported condition types, see the dedicated page:

[Using Conditionals in Messages →](#)

LED Reference

Overview

iReporter can drive hardware LEDs on your iReporter Button Box to provide visual indicators for active crash/incident timers and the Blue Flag Helper approach warning.

Serial Protocol

iReporter communicates with the button box using a simple text-based protocol over the configured COM port at 9600 baud. Each LED command is a single line in the format:

```
LED;{COLOUR};{STATE};
```

Where:

- **COLOUR** — one of: RED, BLUE, GREEN, AMBER
- **STATE** — 1 to turn the LED on, 0 to turn it off

Examples:

```
LED;RED;1;      (turn red LED on)
LED;RED;0;      (turn red LED off)
LED;AMBER;1;    (turn amber LED on)
```

LED Assignments

| Feature | Setting | Behaviour |
|------------------|---|---|
| Crash Capture | Timeout LED (in Crash Capture settings) | Steady ON while crash is active (timed mode) — OR — brief 600ms flash on capture (No Timeout mode) |
| Incident Capture | Timeout LED (in Incident Capture settings) | Steady ON while incident is active (timed mode) — OR — brief 600ms flash on capture (No Timeout mode) |
| Blue Flag Helper | Approach LED (in Blue Flag Helper settings) | ON when a top-X leader is within the configured gap AND more than 1 full lap ahead |

Timed vs No Timeout LED Behaviour

The crash and incident LEDs behave differently depending on the timeout setting:

| Timeout Setting | LED Behaviour | Data Cleared When |
|-----------------|--|---|
| Timed (1-60s) | Stays ON for the full timeout duration, then turns OFF | Timeout expires |
| No Timeout (0) | Flashes ON for 600ms on detection, then turns OFF | Next crash/incident is detected (overwrites previous) |

The brief flash in No Timeout mode confirms that a crash or incident has been captured without implying it is still pending — the data remains available in message variables indefinitely.

Colour Options

| Option | Description |
|--------|--------------------------------|
| None | No LED output for this feature |
| RED | Red LED |
| BLUE | Blue LED |
| GREEN | Green LED |
| AMBER | Amber / orange LED |

Notes

- iReporter only sends a new LED command when the state changes — it does not repeatedly send the same command.
- Multiple features can use different colours simultaneously (e.g. RED for crash, AMBER for incident).
- If two features are assigned the same colour, the last state change wins. It is recommended to assign a unique colour to each feature.
- The COM port and baud rate (9600) are fixed. Ensure your iReporter Button Box firmware listens on 9600 baud and parses the `LED;COLOUR;STATE;` format.

Using Conditional Statements in Messages

Conditional Messages

iReporter supports conditional message lines using a simple bracket syntax. Place a condition in parentheses at the start of any message field — if the condition is false at press time, that line is silently skipped and nothing is sent for it.

Syntax

```
(condition)message text here
```

Everything after the closing `)` is sent only when the condition is true. No closing tag needed — the condition applies to the whole line.

Example

```
(GAPBEHIND1<2)/msg {CARBEHIND1} Car {CAR#} is pitting this lap!
```

This sends a private message to the car immediately behind only if it is within 2 seconds. If the gap is greater the line is skipped entirely and nothing is sent.

AND and OR

Combine multiple conditions on the one line using `&&` (AND) or `||` (OR). AND is evaluated before OR.

| Example | Sends when |
|---|---|
| <code>(CRASH && GAPBEHIND1<5)/rc Contact near traffic!</code> | A crash is active AND car 1 behind is within 5s |
| <code>(CRASH INCIDENT)/rc Incident detected!</code> | A crash OR an incident is currently active |
| <code>(FLAG=Yellow && GAPHEAD<3)/rc Yellow – tight ahead!</code> | Yellow flag AND less than 3s to car ahead |

You can also write `AND` and `OR` in plain English instead of `&&` and `||`.

Available Conditions

Crash and Incident

| Condition | True when |
|------------|-----------------------------------|
| CRASH | A crash is currently detected |
| NOCRASH | No crash is currently active |
| INCIDENT | An incident is currently detected |
| NOINCIDENT | No incident is currently active |

Gap Conditions

Replace **N** with any number of seconds. Operators: **<** **>** **<=** **>=**

| Condition | True when |
|-----------------------------|--|
| GAPHEAD<N or GAPHEAD1<N | Gap to 1st car ahead is less than N seconds |
| GAPHEAD2<N | Gap to 2nd car ahead is less than N seconds |
| GAPHEAD3<N | Gap to 3rd car ahead is less than N seconds |
| GAPBEHIND<N or GAPBEHIND1<N | Gap to 1st car behind is less than N seconds |
| GAPBEHIND2<N | Gap to 2nd car behind is less than N seconds |
| GAPBEHIND3<N | Gap to 3rd car behind is less than N seconds |

Flag Condition

| Condition | True when |
|----------------|-----------------------------------|
| FLAG=Yellow | Current flag is Yellow / Caution |
| FLAG=Green | Current flag is Green (racing) |
| FLAG=Red | Current flag is Red |
| FLAG=White | Current flag is White (final lap) |
| FLAG=Checkered | Current flag is Checkered |

Practical Examples

Notify close cars when pitting

| Field | Content |
|-----------|--|
| Message 1 | <code>/rc Car {CAR#}, {DRIVER} Pitting this LAP</code> |
| Message 2 | <code>(GAPBEHIND1<2)/msg {CARBEHIND1} Car {CAR#} pitting this LAP!</code> |
| Message 3 | <code>(GAPBEHIND2<3)/msg {CARBEHIND2} Car {CAR#} pitting this LAP!</code> |

Message 1 always fires. Messages 2 and 3 fire only when the cars behind are within the gap threshold — each sends a separate private message.

Flag-aware status

| Field | Content |
|-----------|--|
| Message 1 | <code>(FLAG=Yellow)/rc Yellow flag – Car {CAR#} Lap {CARLAP} holding position</code> |
| Message 2 | <code>(FLAG=Green)/rc Racing – Car {CAR#} Lap {CARLAP}</code> |

Combined condition

```
(CRASH && GAPBEHIND1<5)/rc Contact AND close traffic behind – Car {CARBEHIND1} at risk
```

Using the Variable Picker

Type `(` in any message box to open the variable picker and browse all available conditions. AND / OR examples are included at the bottom of the list.

Backward Compatibility

The older `{IF:condition}text{ENDIF}` syntax is still fully supported for any messages you have already saved.