

Notify

Notify adds custom on-screen notifications for players, with a separate style per type, optional sounds, and a chat fallback when a player cannot see popups.

Table of Contents

- Installation
- Quick Start
- Commands
- Permissions
- Configuration
- Features
- Localization
- Integrations
- API
- Offline Image Mode
- Troubleshooting
- Test Server
- Support

Installation

1. Download `Notify.cs` and place it in `oxide/plugins` (or `carbon/plugins` for Carbon)
2. Reload the plugin: `o.reload Notify` (Oxide) or `c.reload Notify` (Carbon)
3. Configure the plugin via the generated config file at `oxide/config/Notify.json`
- Install the ImageLibrary plugin (required on Oxide for images, unless you use [Offline Image Mode](#))

Quick Start

1. Grant `notify.see` to players who should see popups. Grant `notify.notify` to yourself.
2. In chat, run `notify.show 0 Welcome`.
3. A popup titled Notification (English) appears at the top right and stays for Notify Cooldown seconds (10 by default).
4. Edit Notifications (type - settings) in `oxide/config/Notify.json` to restyle that type or add a new type number.
5. Reload the plugin.

Commands

These commands are registered for chat and for the server console. There is no command that is limited to the server console, and there is no editor command. A chat caller with the wrong permission gets no reply. The server console skips the permission check.

The on-screen close button runs `notify.close`. That is not a command to type. It is documented under Permissions and Troubleshooting because it checks `notify.notify`.

Player Commands

Ordinary players do not type a command to open Notify. Popups are sent by the admin commands below or by another plugin through the API.

Admin Commands

- `notify.show <type> <message>` — show that type to yourself. Requires `notify.notify`. The message is every word after the type. Fewer than two words, or a type that is not a whole number, gets a syntax reply. The server console passes the permission check and then does nothing, because there is no player to attach the popup to.

- `notify.player <name or id> <type> <message>` — show that type to one player. Requires `notify.player`. The player is matched by name or id and must be online. If nobody matches, the command replies that the player was not found. The server console may run this with no permission.
- `notify.allplayers <type> <message>` — show that type to every connected player. Requires `notify.allplayer`. The permission text is `notify.allplayer`. The command text is `notify.allplayers`. Fewer than two words, or a type that is not a whole number, gets a syntax reply. The server console may run this with no permission.

The target still needs `notify.see`, or the popup is replaced by chat or dropped. See Permissions. If the type number is missing from the config, or that type has Enabled set to false, the command finishes and nothing is drawn.

Permissions

- `notify.see` — see an on-screen popup. Without it, `SendNotify` does not draw. When `Send text message to chat if player doesn't have notification permission` is true, the message is written in chat instead. When that option is false, the message is dropped.
- `notify.notify` — run `notify.show` from chat. The close button runs `notify.close` and checks this same permission, so a player who only has `notify.see` can see a popup and cannot dismiss it with the button. It still disappears when its cooldown ends. Grant `notify.notify` to anyone who should dismiss popups. That also lets them run `notify.show`.
- `notify.player` — run `notify.player` from chat.
- `notify.allplayer` — run `notify.allplayers` from chat. Do not invent a fourth permission named like the command. The registered string is `notify.allplayer`.

Configuration

Settings are written to `oxide/config/Notify.json` (Or `carbon/configs/Notify.json` on Carbon). Leave the Version entry in the file. The plugin updates Version when it migrates an older file. If the file cannot be parsed, the plugin prints `Your configuration file contains an error. Using default configuration values.`, keeps running on defaults in memory, and leaves the broken file on disk. Fix the file, then reload.

While upgrading an older config, a stored display type of `OverlayNonScaled` is rewritten to `Overall`, so popups show on top of plugin windows and modal windows. A file older than 1.0.14 that stores `Overlay` is rewritten to `Overall` as well. After that upgrade, the value you set is kept. Updating to 1.0.20 also replaces the old default award image addresses (`i.ibb.co`) with the `PluginsStorage` ones; your own image addresses are kept.

Display type (Overlay/Hud)

This value is the parent layer of the popup panel. Use `Overall`, `Overlay`, `Hud`, or `OverlayNonScaled`. The default is `Overall`, which draws popups on top of plugin windows and modal windows. A parent the game does not draw will hide every popup.

`Height` and `Width` are the default popup size in pixels. A type can override them. `X Margin` is the gap from the side of the screen. `Y Margin` is the gap between stacked popups. `Y Indent` is the vertical start of the stack. `Display notifications on the top right?` uses the top right corner when true. The default `Y Indent` of `-50` then places the first popup just below that corner. When the switch is false, the anchor is the bottom left corner and the same offset is applied, so a negative `Y Indent` sits below the screen. Use a positive `Y Indent` at least as tall as the popup for a bottom-left stack.

```
{
  "Display type (Overlay/Hud)": "Overall",
  "Height": 50,
  "Width": 260,
  "X Margin": 20,
  "Y Margin": 5,
  "Y Indent": -50,
  "Display notifications on the top right?": true
}
```

Notify Cooldown

`Notify Cooldown` is the number of seconds the oldest visible popup stays up, unless that type sets `Use custom cooldown`. `Max Notifications On Screen` is how many popups are drawn at once, from the front of the queue. Only the oldest popup's timer runs. When it ends, that popup is removed and the next one starts its own cooldown. Popups past the max stay queued and are drawn as earlier ones leave.

`Send text message to chat if player doesn't have notification permission` is the chat fallback for players who lack `notify.see`. `Enable Offline Image Mode` is covered in `Offline Image Mode`.

```
{
  "Notify Cooldown": 10,
  "Max Notifications On Screen": 10,
  "Send text message to chat if player doesn't have notification permission": true,
  "Enable Offline Image Mode": false
}
```

Notifications (type - settings)

Each key is a whole number. Commands and `SendNotify` pass that number as the type. The title is not the message. The title is the language string named by `Title Key (lang)`. The message is the body text passed to the command or the API.

Colors are an object with `HEX` and `Opacity (0 - 100)`. `Align` is a name such as `MiddleCenter`, `LowerLeft`, `UpperLeft`, `LowerRight`, or `UpperRight`. `fontSize` and `Is Bold?` style that text block.

`Background Image` empty draws a flat panel in `Background Color`. A non-empty value draws that image tinted by `Background Color`. `Enable Gradient?` paints `Gradient Color` across the popup with `Sprite` and `Material`.

`Image Settings` with `Enabled` true and a non-empty `Image` replaces the icon. Otherwise the icon is a square in `Icon Color` with `Icon Text` in the middle.

`Use custom width` and `Use custom height` switch that type from the global `Width` and `Height` to `Custom width` and `Custom height`. `Use custom cooldown` switches that type from `Notify Cooldown` to its own `Cooldown`.

`Use command` makes the whole popup run `Command` when clicked. `Close after using the command?` also tells the client to close that popup element. It does not remove the popup from the queue, so a later redraw can show it again. The close button does remove it from the queue.

`Sound Effect` (empty - disable) plays for that player when the popup is added. Empty disables it. `Fade In` and `Fade Out` are seconds. `Close Button Settings` with `Enabled` false hides the button. `Close text` is the button label.

Type `0` is the default info style. The title key is `Notification`. It uses the global size and the global cooldown. The icon text is `!`.

```
{
  "0": {
    "Enabled": true,
    "Background Image": "",
    "Background Color": {
      "HEX": "#000000",
      "Opacity (0 - 100)": 98
    },
    "Enable Gradient?": true,
    "Gradient Color": {
      "HEX": "#4B68FF",
      "Opacity (0 - 100)": 35
    },
    "Sprite": "assets/content/ui/ui.background.transparent.linearltr.tga",
    "Material": "Assets/Icons/IconMaterial.mat",
    "Icon Color": {
      "HEX": "#4B68FF",
      "Opacity (0 - 100)": 100
    },
    "Icon Text": "!",
  }
}
```

```

    "Title Key (lang)": "Notification",
    "Fade Out": 1,
    "Fade In": 0.1,
    "Sound Effect (empty - disable)": "assets/bundled/prefabs/fx/notice/item.select.fx.prefab",
    "Image Settings": {
        "AnchorMin": "0 0",
        "AnchorMax": "0 0",
        "OffsetMin": "12.5 12.5",
        "OffsetMax": "37.5 37.5",
        "Enabled": false,
        "Image": ""
    },
    "Use custom width": false,
    "Custom width": 0,
    "Use custom height": false,
    "Custom height": 0,
    "Use command": false,
    "Command": "",
    "Close after using the command?": false,
    "Close Button Settings": {
        "AnchorMin": "1 1",
        "AnchorMax": "1 1",
        "OffsetMin": "-25 -25",
        "OffsetMax": "-1 -1",
        "Enabled": true,
        "Close text": "x",
        "fontSize": 12,
        "Is Bold?": false,
        "Align": "MiddleCenter",
        "Color": {
            "HEX": "#FFFFFF",
            "Opacity (0 - 100)": 60
        }
    },
    "Icon Settings": {
        "AnchorMin": "0 0.5",
        "AnchorMax": "0 0.5",
        "OffsetMin": "12.5 -12.5",
        "OffsetMax": "37.5 12.5",
        "fontSize": 12,
        "Is Bold?": false,
        "Align": "MiddleCenter",
        "Color": {
            "HEX": "#FFFFFF",
            "Opacity (0 - 100)": 100
        }
    },
    "Title Settings": {
        "AnchorMin": "0 0.5",
        "AnchorMax": "1 1",
        "OffsetMin": "47.5 0",
        "OffsetMax": "0 0",
        "fontSize": 12,
        "Is Bold?": false,
        "Align": "LowerLeft",
        "Color": {
            "HEX": "#FFFFFF",
            "Opacity (0 - 100)": 50
        }
    },
    "Text Settings": {
        "AnchorMin": "0 0",
        "AnchorMax": "1 0.5",
        "OffsetMin": "47.5 0",
        "OffsetMax": "0 0",
        "fontSize": 10,
        "Is Bold?": false,

```

```

    "Align": "UpperLeft",
    "Color": {
      "HEX": "#FFFFFF",
      "Opacity (0 - 100)": 100
    }
  },
  "Use custom cooldown": false,
  "Cooldown": 0
}
}

```

Other default types:

- `1` — title key `Error`, icon text `x`, red gradient, global size, global cooldown, no click command.
- `2130354` — title key `AwardAvailable`, custom size 280 by 60, click runs `daily` and closes the element, custom cooldown 60, background image and icon image set.
- `2130355` — title key `AwardCooldown`, custom size 280 by 60, click runs `daily` and closes the element, custom cooldown 0.9.
- `2130356` — title key `AwardReceived`, custom size 280 by 60, click runs `daily` and closes the element, custom cooldown 60.

Copy an entry and change the number to add a type. Set `Enabled` to false to keep the entry but stop drawing it.

Features

On-screen popups

Each notice is a small panel with a title, a message, an icon or an image, an optional gradient, and an optional close button. Several popups stack in one corner.

Separate types

The type number picks the style. Types `0` and `1` are a general notice and an error. Types `2130354`, `2130355`, and `2130356` are award layouts. Unknown numbers and disabled types are ignored.

Queue and cooldown

The oldest popup controls the timer. Newer ones wait, and only `Max Notifications On Screen` are drawn. A type may use the global cooldown or its own.

Click action

A type can run a command when the player clicks the popup. The default award types run `daily`. Closing the element on click does not delete the queued notice. The close button does.

Sound

A type can play a sound effect for that player when the notice is added. An empty effect path plays nothing.

Chat fallback

Players without `notify.see` can receive the same text in chat, or nothing, depending on the config switch.

Images

Backgrounds and icons can be remote images or files under the `TheMevent` data folder. On Oxide, remote images go through ImageLibrary. On Carbon, they go through the built-in image database. Text-only types such as `0` and `1` still draw when no image plugin is loaded. The default award types use images from the Mevent PluginsStorage. On Oxide without ImageLibrary they are skipped and

`IMAGE LIBRARY IS NOT INSTALLED.` is printed five times, unless Offline Image Mode is on and the images from the archive are in the data folder.

Languages

English and Russian strings are included. Popup titles are language keys, so the title can differ from the message body.

Localization

Messages are registered for English and for Russian. After the first load, edit the generated files:

- `oxide/lang/en/Notify.json` OR `carbon/lang/en/Notify.json`
- `oxide/lang/ru/Notify.json` OR `carbon/lang/ru/Notify.json`

Title Key (lang) must match a key in that file. A missing key is shown as the key itself. Add a new key there when you invent a title.

Key	English	Russian
Notification	Notification	Оповещение
Error	Error	Ошибка
AwardAvailable	NEW AWARD AVAILABLE!	ДОСТУПНА НОВАЯ НАГРАДА!
AwardCooldown	AWARD WILL BE AVAILABLE	НАГРАДА БУДЕТ ДОСТУПНА
AwardReceived	REWARD RECEIVED	НАГРАДА ПОЛУЧЕНА
SyntaxNotify	Error syntax! Use: <code>/{} [type] [message]</code>	Ошибка синтаксиса! Используйте: <code>/{} [type] [message]</code>
SyntaxPlayerNotify	Error syntax! Use: <code>/{} [steamid] [type] [message]</code>	Ошибка синтаксиса! Используйте: <code>/{} [steamid] [type] [message]</code>
SyntaxAllPlayerNotify	Error syntax! Use: <code>/{} [type] [message]</code>	Ошибка синтаксиса! Используйте: <code>/{} [type] [message]</code>
NotFound	Player ' <code>{}'</code> not found!	Игрок ' <code>{}'</code> не найден!

`{}` in the syntax lines is the command text the player typed. `{}` in NotFound is the name or id that did not match.

Integrations

Other Plugins

- **ImageLibrary** — required on Oxide for remote images. Notify imports each background and each enabled icon image. If ImageLibrary is missing or not loaded while any remote image is configured, the plugin prints `IMAGE LIBRARY IS NOT INSTALLED.` five times and skips the import. Carbon does not use that plugin. Carbon queues the same images through its built-in image database. With Offline Image Mode on, PluginsStorage images are read from the data folder and need neither.
- Other plugins on the server call `SendNotify` or `SendNotifyAllPlayers` to push a popup, and `TryToggleType` to turn a type on or off. This file does not name those plugins. The type number they pass must exist under `Notifications (type - settings)`.

API

Hooks Called

This plugin does not raise an On or Can hook for other plugins to subscribe to.

API Methods

- `SendNotify(BasePlayer player, int type, string message)` — queue a popup for that player. Does nothing if the player is missing. If the player lacks `notify.see`, the chat fallback rules apply and no popup is drawn. If the type is missing or disabled, nothing is drawn.
- `SendNotify(string userId, int type, string message)` — same rules. The id string must be numeric. If that number matches no player, nothing is sent.
- `SendNotify(ulong userId, int type, string message)` — same, with a numeric id.

- `SendNotifyAllPlayers(int type, string message)` — calls `SendNotify` for every connected player. Each player is checked for `notify.see` on their own.
- `TryToggleType(int type, bool newValue)` — sets `Enabled` on that type and saves the config. Returns false when the type does not exist or `Enabled` is already `newValue`.

Offline Image Mode

The plugin's default images are included in the archive, so the server does not have to download them:

1. Copy the `data` folder from the archive into `oxide/` (or `carbon/` on Carbon). The images end up in `oxide/data/TheMevent/Images/Notify/`.
2. Set "Enable Offline Image Mode" to `true` in `oxide/config/Notify.json` (or `carbon/configs/Notify.json`).
3. Reload the plugin.

With the switch on, every default image is read from `data/TheMevent/` instead of being downloaded, and `ImageLibrary` is not needed for them. A file that is missing there is reported in the console as

`Image not found: TheMevent/...`

For your own images, put the file under `oxide/data/TheMevent/` and set the image value to `TheMevent/<path>`, for example `"TheMevent/MyImage.png"`. Such values are read from the data folder even when the switch is off. Other web addresses are always downloaded.

Troubleshooting

Images not loading (HTTP 429 errors):

- Enable Offline Image Mode (see above)
- Or use `ImageLibrary` plugin

No popup:

- The player needs `notify.see`. If the chat fallback is on, read chat instead.
- The type number must exist under `Notifications (type - settings)`, and `Enabled` must be true.
- `notify.show` from the server console has no player, so it never draws.
- A chat command with the wrong permission returns without a message.
- `Display type (Overlay/Hud)` must be a layer the game draws, such as `Overall`, `Overlay`, `Hud`, or `OverlayNonScaled`.

Close button does nothing:

- The button runs `notify.close`, which requires `notify.notify`.
- The popup still leaves when its cooldown ends.
- `Close Button Settings` must have `Enabled` set to true.

Broadcast permission does not match the command:

- The command is `notify.allplayers`.
- The permission is `notify.allplayer`.

Remote images missing on Oxide:

- The console prints `IMAGE LIBRARY IS NOT INSTALLED.` five times when a configured image still needs a download and `ImageLibrary` is not loaded.
- Types `0` and `1` do not need an image. The default award types do.
- On Carbon, a missing `ImageLibrary` plugin does not print that line. Carbon uses its built-in image database.
- With Offline Image Mode on and the archive's `data` folder copied, the default award images do not need `ImageLibrary`.

Clicked award popup comes back:

- `Close after using the command?` only closes the element. It leaves the notice in the queue.
- Use the close button, which requires `notify.notify`, or wait for the cooldown.

Config will not load:

- A broken file is not overwritten. The plugin prints
Your configuration file contains an error. Using default configuration values. and uses defaults until the file is fixed and the plugin is reloaded.

Test Server

Join our test server to try Notify:

```
connect 155.103.86.38:28015
```

Support

- Discord: <https://discord.gg/mevent>