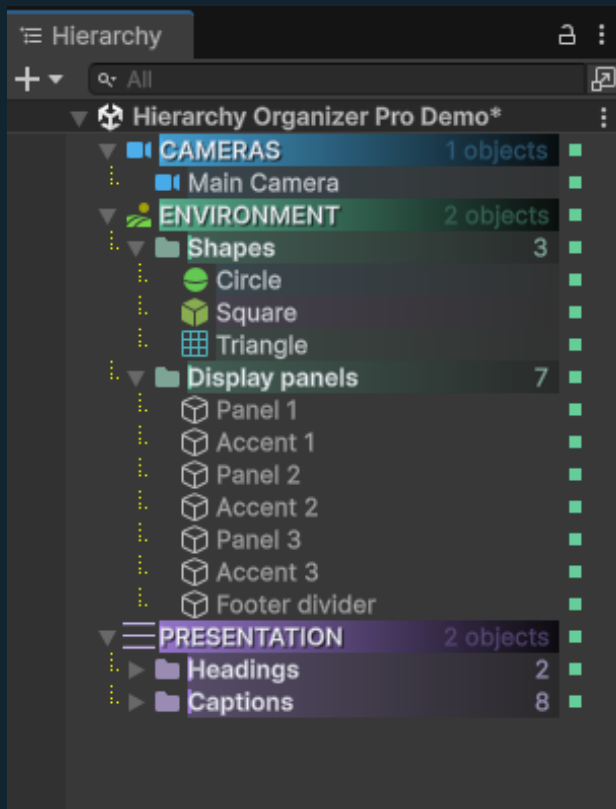


Hierarchy Organizer Pro

The complete guide from the beginning

Organize your Hierarchy with color, typography and icons.
Learn every window, reusable styles, prefab inheritance,
scene navigation and configuration - step by step.



01

QUICK EDITING

Style rows in a few clicks.

02

ADVANCED CONTROL

Tune colors, icons and groups.

03

REUSABLE WORKFLOWS

Save styles and share settings.

START HERE

How to use this guide

Start with one existing GameObject and follow the first styling walkthrough. Then use the control tables to tune your workflow. Screenshots show the interface supplied by ERAGANX; the code defines the behavior explained here.

Reading path - click to jump	Pages
Foundation, installation and your first styled row	03 - 05
Settings, gestures and the Quick Window	06 - 09
Advanced editing, colors, icons and row details	10 - 16
Profiles, reusable styles and group commands	17 - 20
Prefabs, Hierarchy features and scene navigation	21 - 24
Exports, imports and persistent project data	25 - 27
A complete scene workflow and Editor C#	28 - 30
Troubleshooting, checklist and FAQ	31 - 33
Glossary, sources and version scope	34

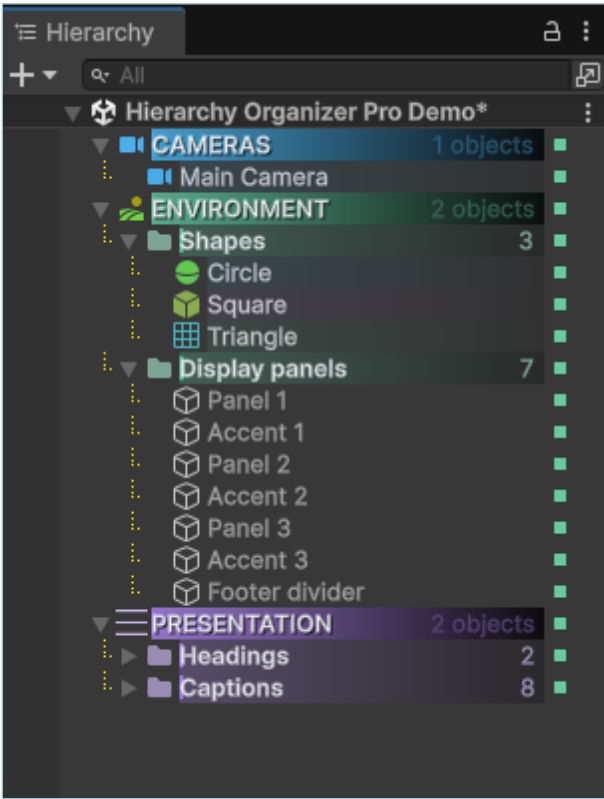
Choose your starting point

First use: pages 4-5. **Fast row edits:** pages 8-9. **Custom reusable styles:** pages 17-19. **Prefab consistency:** page 21. **Backups or another project:** pages 25-27.

Two kinds of saving
Quick and Advanced edits apply immediately. A reusable style edited in Settings is a draft until you save it. Save your scene to give scene objects durable identities; preserve the project metadata as well.

01 - FOUNDATION

What this package does



A supplied example: stronger Title gradients, softer group rows, icons, counts and optional activation indicators.

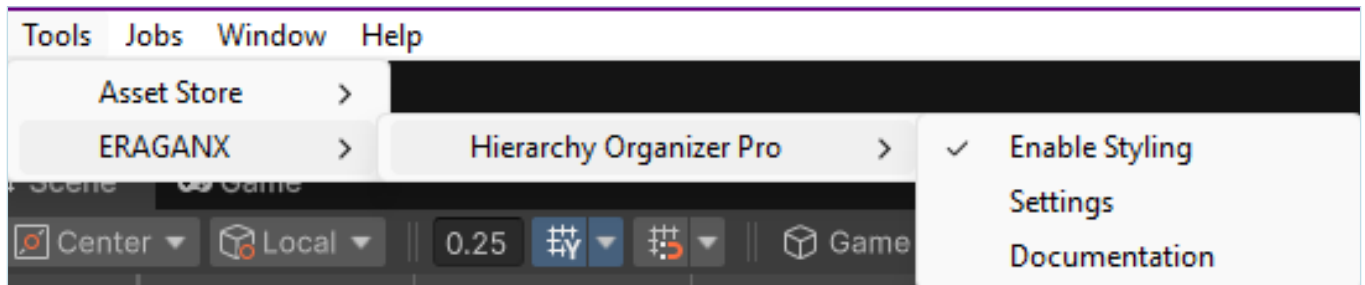
Control / choice	Purpose / action
Presentation	Colors and horizontal gradients; bold, italic, uppercase and alignment; native-height Title and Subtitle roles.
Organization	Reusable headers/subheaders, explicit child colors, object counts and metadata notes.
Editor utilities	Optional activation controls, component mini editing, local tree connectors and a compact scene bookmark bar.
Data	Styles and assignments are stored in ProjectSettings. Configuration and preset JSON support backup and sharing.

Understand the scope

Styling changes Editor presentation without renaming, reparenting or adding components to objects. Activation and component-property editing explicitly change GameObjects. Hierarchy styling does not appear in a built Player.

01 - INSTALLATION

01. Install and open the tools



Tools > ERAGANX > Hierarchy Organizer Pro. The menu has three entries.

- 1 Open your Unity project. This guide matches source targeting **Unity 6000.3.13f1**. Verify other Editor versions in your own project.
- 2 Import or retain **Assets/ERAGANX/Hierarchy Organizer Pro/**, including Editor, UI resources, Documentation and Licenses. Wait for script compilation to finish.
- 3 Resolve Console compile errors, then open **Tools > ERAGANX > Hierarchy Organizer Pro > Settings**.
- 4 Turn on **Enable styling**. In **Window & Interaction**, select Quick Window and the default Alt Click gesture.
- 5 Work in Edit Mode. Save the scene before expecting its assignments to survive closing and reopening the project.

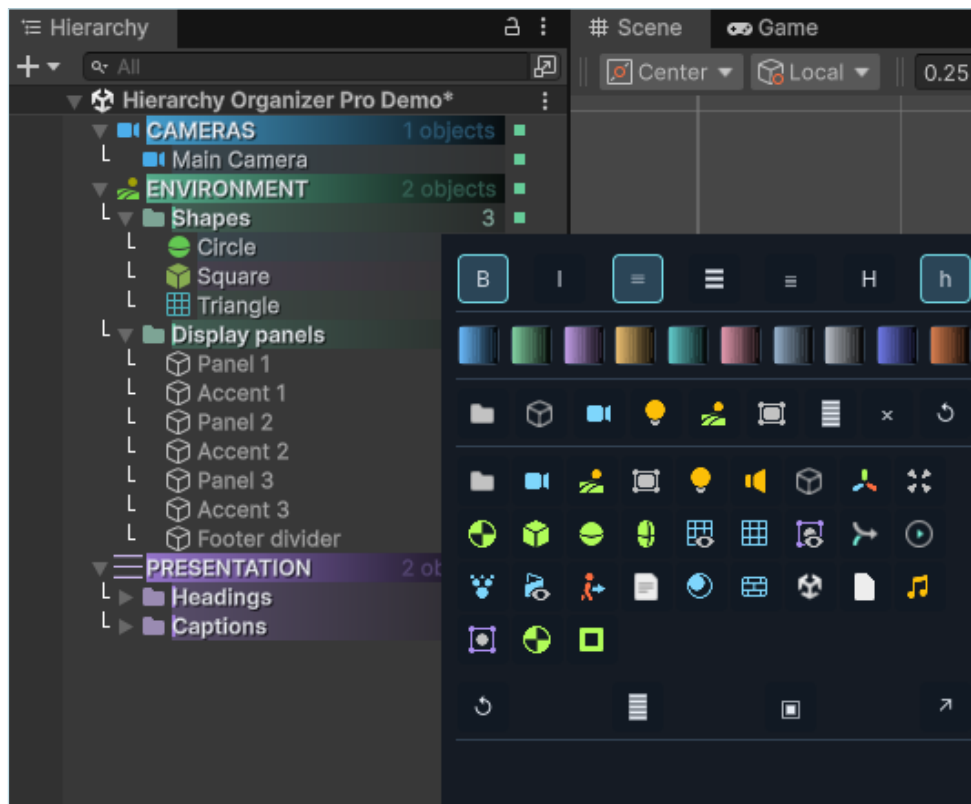
Control / choice	Purpose / action
Enable Styling	Toggles the global styling display. A check mark shows that it is enabled.
Settings	Opens the main controller for interaction, profiles, features, icons and data.
Documentation	The current source opens Documentation/README.md. This complete PDF is at the package root.

No scene manager is required

You style existing hierarchy objects. This Editor assembly has no URP dependency and does not require a camera, shader, runtime script or manager component for styling.

02 - FIRST WALKTHROUGH

02. Style your first group



Alt-click opens the chosen editing window beside a hierarchy row.

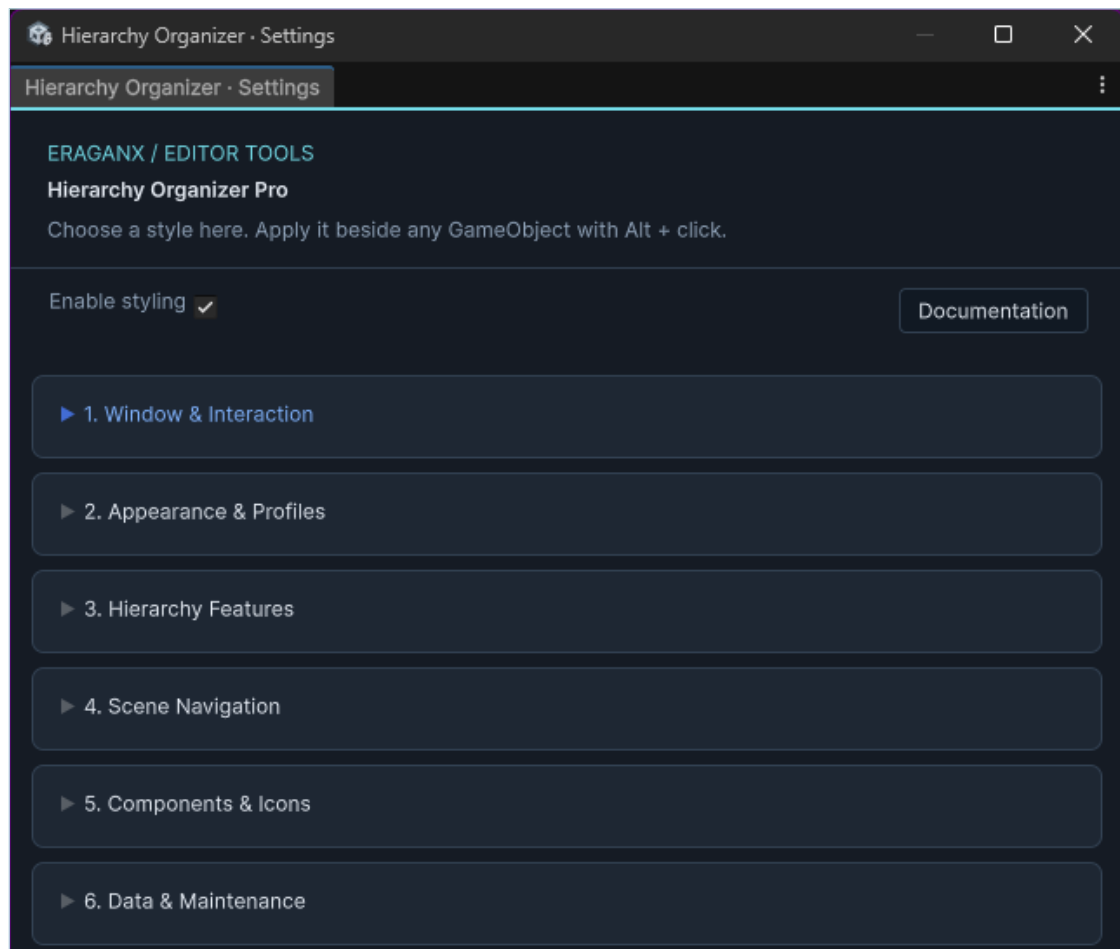
- 1 Choose an existing parent such as CAMERAS. If you want only this row, select it alone, then **Alt + left click its object row**. Avoid the foldout arrow.
- 2 Click **H** to apply the Title profile. Pick a gradient color. The hierarchy displays the actual GameObject name.
- 3 Choose a camera or folder icon. Use Bold and alignment controls to make the group easy to identify.
- 4 Click outside to finish. Select a subgroup and use **h** for Subtitle, then choose a softer color.
- 5 Use the Quick Window arrow to open Advanced for counts or child colors. Save your scene, and use Unity Undo if you want to reverse an edit.

Selection matters

When the clicked object is already selected, the editing panel can target the selected editable GameObjects together. Check the selection before applying a profile, preset or reset.

02 - SETTINGS

03. Understand the main controller



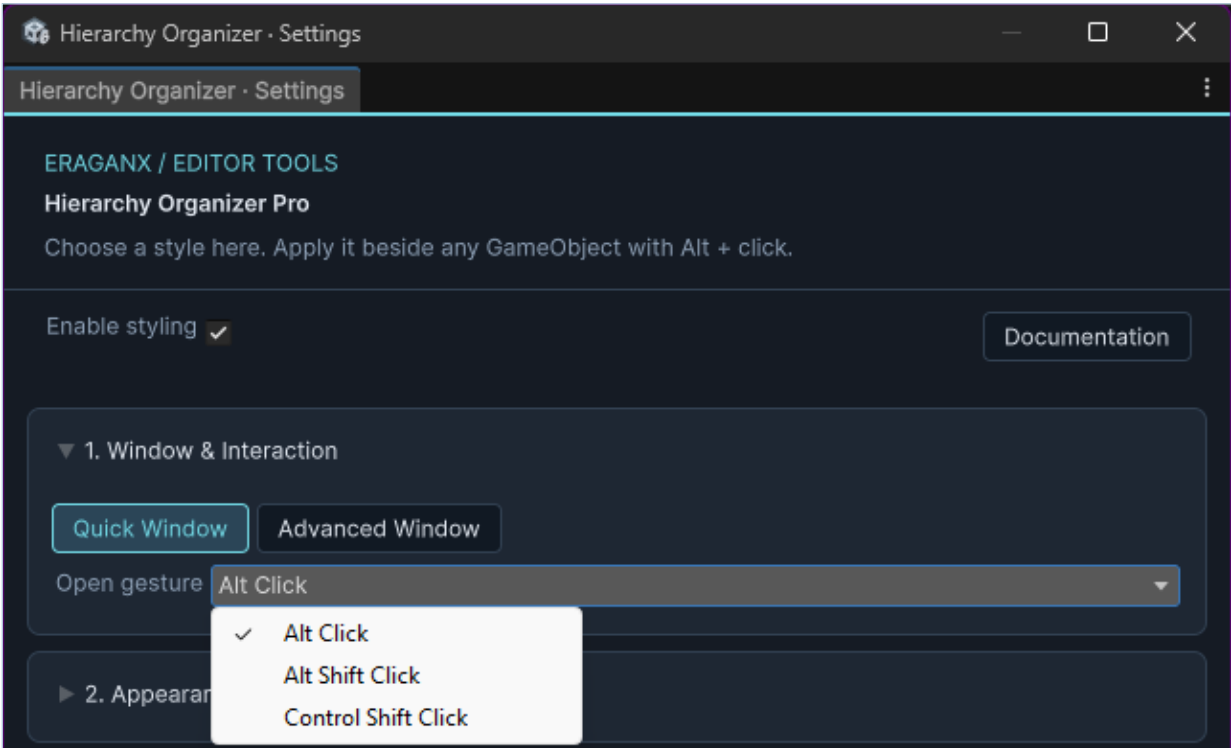
The Settings window groups global controls into six expandable sections.

Control / choice	Purpose / action
1. Window & Interaction	Choose Quick/Advanced and the opening gesture.
2. Appearance & Profiles	Set role profiles, ten Quick colors, font sizes and reusable styles.
3. Hierarchy Features	Activation, tree lines, collapse and optional global defaults.
4. Scene Navigation	Enable the scene bar and add bookmarks.
5. Components & Icons	Component Minimap, Icon package and imported images.
6. Data & Maintenance	Preset/configuration transfer and extra right-side space.

Collapsing a section only hides its controls. Enable styling and Documentation stay at the top; scroll within the window to reach controls below the visible area.

02 - WINDOW & INTERACTION

04. Choose your window and gesture



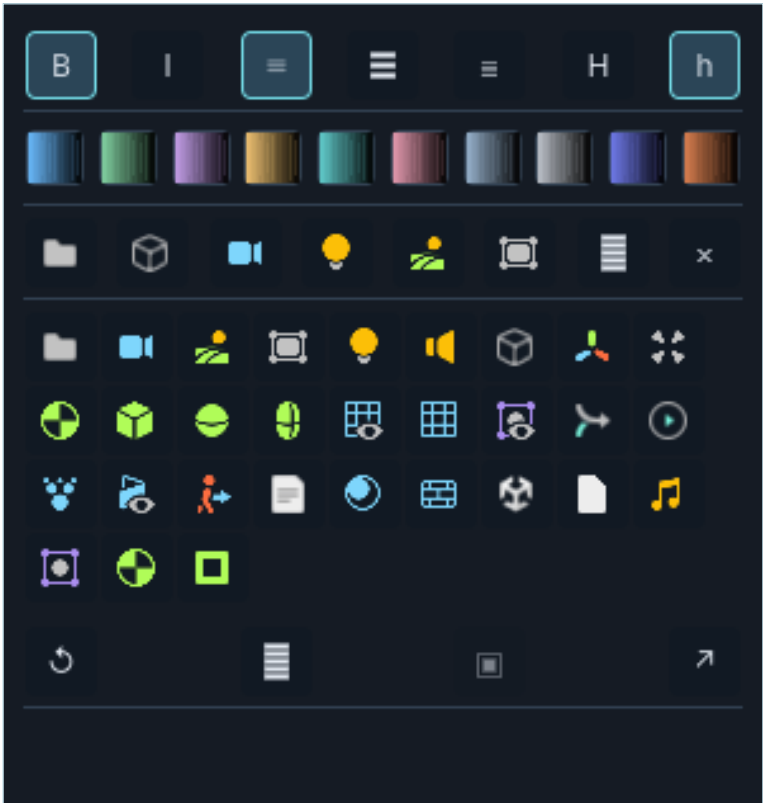
Choose a default window and a gesture that fits your Editor setup.

Control / choice	Purpose / action
Quick Window	Compact formatting, ten gradient colors, thirty Unity icons and row actions.
Advanced Window	Detailed colors, fonts, icon browser, presets, child colors, counts and metadata.
Alt Click	Default: Alt + left click on the object row.
Alt Shift Click	Alternative: Alt + Shift + left click.
Control Shift Click	Alternative: Control/Command + Shift + left click, depending on the platform.

Keep normal foldout behavior

The opening gesture applies over the object row. Native foldout arrows retain Unity behavior. If another extension consumes Alt-click, change the gesture here. Row styling is disabled during Play Mode.

05. Learn the compact controls



The expanded icon palette is shown. Hover each button for its description.

Control / choice	Purpose / action
Formatting row	B: Bold. I: Italic. Three alignment buttons: left, center, right. H: Title. h: Subtitle.
Gradient row	Exactly ten colors. Each fills the row with a horizontal gradient toward a dark version of the color.
Icon row and expander	Six common choices, an expander for all thirty Unity icons, No icon and Restore Unity icon.
Action row	Reset style; Copy; Paste; Open Advanced Window.

Every click is an edit

There is no Apply button in Quick. Outside input or Escape closes the popup with edits retained. Reopen the row to continue; Undo restores earlier changes. Use Advanced to apply named reusable styles.

03 - QUICK WORKFLOW

06. Profiles, color and row actions

A repeatable order

- 1

Select the intended row or rows. Apply **H** for a major Title or **h** for a Subtitle.
- 2

Choose a Quick gradient color, then adjust Bold, Italic or alignment. Applying H/h copies the corresponding global profile, so apply the role before final custom formatting.
- 3

Pick an icon. Expand the palette for the full set. Click No icon to hide the glyph, or Restore Unity icon to return to the native appearance.
- 4

Close the popup. Reopen it and use the arrow to Advanced whenever you need precise strength, a custom color, counts or saved style cards.

Control / choice	Purpose / action
Copy	Copies the first target's resolved style into the system clipboard as a versioned package payload.
Paste	Applies the copied style to target rows. It can also apply the copied style's explicit child-color recipe.
Reset	Removes the selected rows' own assignments. A prefab source or global default can become visible again.
Invalid clipboard	Paste is disabled when the clipboard has no valid Hierarchy Organizer Pro style.
Multiple rows	Property edits keep each row's unrelated choices. A profile, preset or paste applies a complete style snapshot.

Example choices

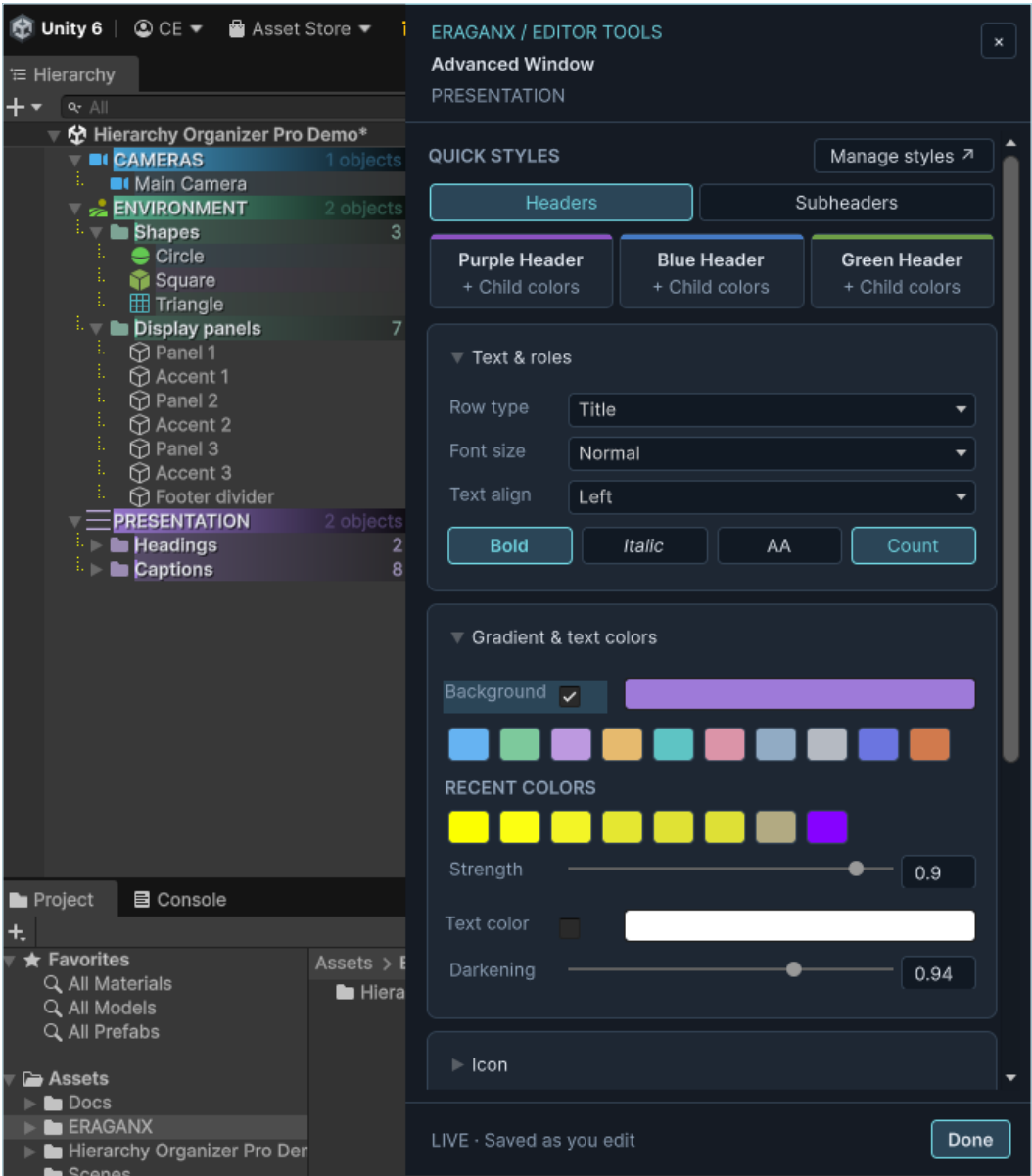
Example object	Suggested workflow
CAMERAS	Title, blue, bold, left aligned, camera icon.
ENVIRONMENT	Title, green; later add soft descendant colors in Advanced.
Shapes / Display panels	Subtitle or custom subheader, subtle gradient, matching folder icon.

Roles do not create text fields

Title and Subtitle format the real object name. They keep Unity's native row height. Optional descriptions and notes are shown through tooltips.

04 - ADVANCED WINDOW

07. Open the detailed editor



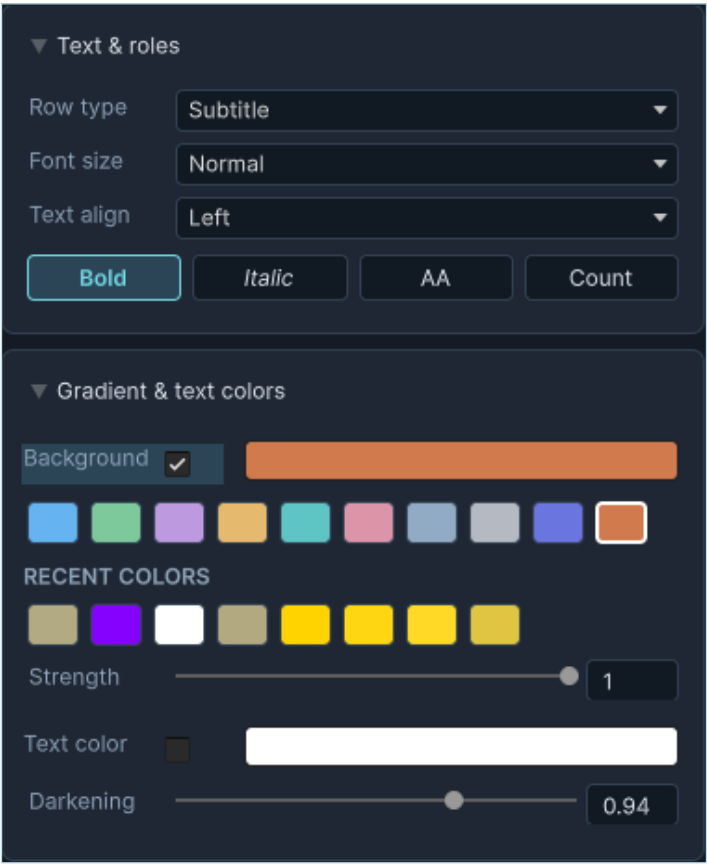
Advanced combines style cards with collapsible editing sections. Changes are saved as you edit.

Choose Advanced in Settings or use the Quick Window arrow. The top identifies the clicked object or selection. **Headers** and **Subheaders** choose which reusable cards are shown; **Manage styles** opens Settings.

Control / choice	Purpose / action
Live row editing	Edit text, colors, icons and details immediately. Done or outside focus finishes the panel.
Collapsible sections	Expand only what you need and scroll for child colors, counts, notes and group commands.

04 - TEXT & ROLES

08. Format names without changing them



A detail view of Text & roles and Gradient & text colors in Advanced.

Control / choice	Purpose / action
Row type	Object: ordinary row. Title: section role. Subtitle: supporting role. The displayed text remains the object name.
Font size	Small, Normal or Emphasized. Numeric sizes are configured only in Settings; all default to Unity's size.
Text align	Left, Center or Right within the available title area.
Bold / Italic / AA	Toggle emphasis or display uppercase. AA changes presentation, not the object's actual name.
Count	Shows the direct child count. Further counting controls appear in Counts & details.

Native height and available width

Rows keep Unity's fixed height. Extra indicators reduce the width available for names; long names use clipping and tooltips. Subtitle is a role, not a second line of hierarchy text.

04 - GRADIENT & TEXT COLORS

09. Tune color, strength and darkening

The background is a horizontal color-to-dark gradient. Use a strong fill for major groups and a lighter fill for supporting rows. Text color is independent of the background.

Control / choice	Purpose / action
Background	Enables or disables the fill. Picking a background color enables it automatically.
Color and swatches	Use any custom color, the ten swatches or a recent color. The ten configurable Quick colors are managed in Settings.
Strength	0-1 background opacity. 0 is invisible; 1 gives the strongest fill. This is separate from gradient darkening.
Darkening	0.8-1. Controls how dark the far end becomes. At 0.94, the end uses about 6% of the base RGB values.
Text color	Enable the switch or choose a color to enable it. The override also applies when the row is selected.
Recent colors	The last eight remembered colors are kept for convenient reuse.
Settings picker	Settings uses Unity's native color picker. Advanced uses an owned HSV/Hex picker.

A practical color recipe

- 1

Start with blue, green or purple. Use a stronger Title fill and a lower-strength Subtitle fill.
- 2

For descendants, use the same color with a subtle strength such as **0.18-0.24**. This is a starting point, not a required value.
- 3

Check legibility for selected, unselected and inactive rows. Keep text bright enough against the dark gradient.
- 4

Keep the pointer in the owned picker while dragging. Done or Escape returns to Advanced; outside Editor focus finishes editing.

Quick versus detailed swatches

Settings' ten gradient presets feed the compact Quick Window. The reusable-style draft and Advanced swatch rows use the source's built-in palette. Use custom or recent colors there when you need an exact configured Quick color.

04 - ICONS

10. Choose, hide or restore an icon

Quick provides thirty Unity choices. Advanced adds common and recent icons, a searchable browser and icon tint. All custom libraries use the single **Icon package** switch in Settings.

Control / choice	Purpose / action
Common / recent	Click to apply an icon immediately. Common contains twelve choices; recent history retains up to twelve icons.
Browse icons	Choose Unity, Google Material, Imported or Project images. Search within the active library; use Previous/Next for pages.
Project images	Choose a Texture2D or Sprite from Assets. The importer creates a private metadata copy; Refresh project images updates the list.
No icon	Hides the object glyph, including a prefab glyph, while retaining other styling.
Unity icon	Restores the native glyph. This action and No icon remain available even with Icon package disabled.
Tint	Tints the chosen custom icon. Select an icon before judging the effect.

The thirty Quick choices

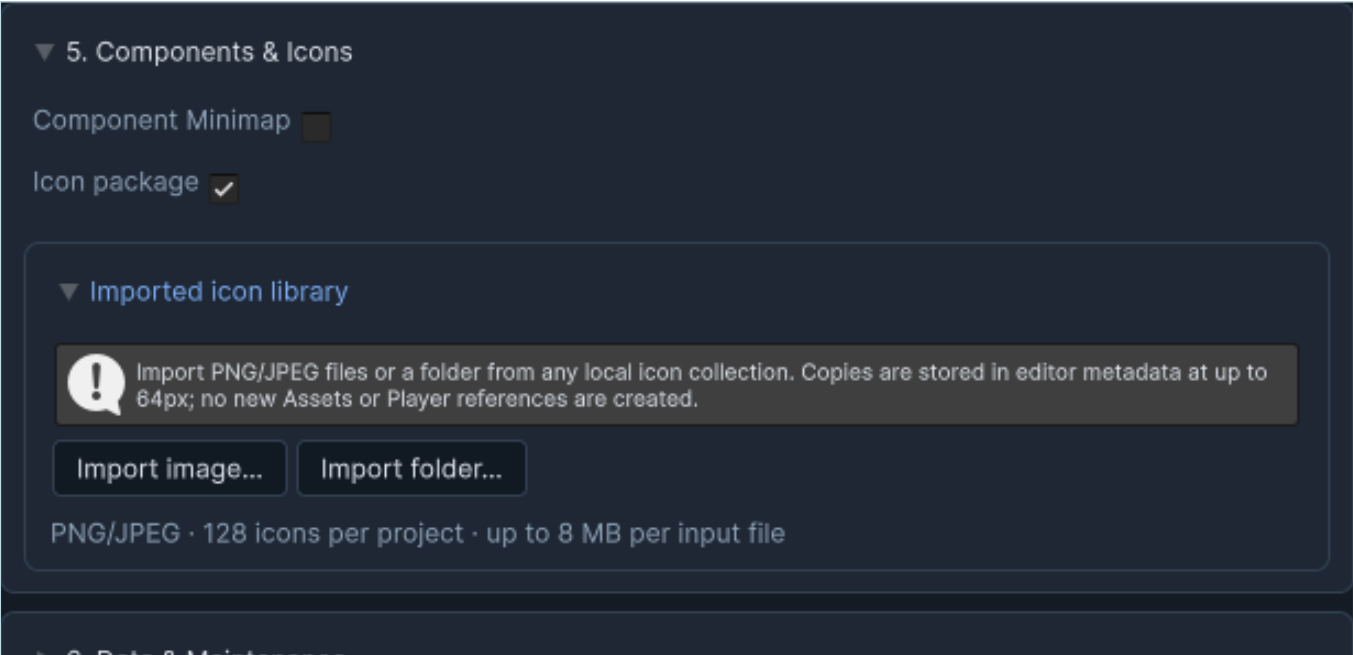
Category	Icon names
Groups and content	Folder, GameObject, Transform, RectTransform, SceneAsset, ScriptableObject, TextAsset.
Rendering and UI	Camera, Terrain, Canvas, Light, MeshRenderer, MeshFilter, SpriteRenderer, SkinnedMeshRenderer, Material, Texture2D, Sprite.
Physics and navigation	Rigidbody, BoxCollider, SphereCollider, CapsuleCollider, Rigidbody2D, BoxCollider2D, NavMeshAgent.
Animation, audio, effects	Animator, Animation, AudioSource, AudioClip, ParticleSystem.

Designed for the native icon slot

Icons fit Unity's 16-point slot and preserve aspect ratio. Private imported, Google and curated images trim transparent padding during cached preparation. Artwork with broad padding may need a tighter source image.

04 - IMPORTED ICON LIBRARY

11. Bring your own PNG or JPEG icons



Settings > Components & Icons > Imported icon library.

- 1 Enable **Icon package**. Click **Import image...** for one file, or **Import folder...** for a local folder.
- 2 Use PNG, JPG or JPEG. Folder import scans files in the selected folder; it does not recursively import subfolders.
- 3 Open Advanced > Icon > Browse icons > Imported and click your image. A Settings import thumbnail can also put it into the current style draft.

Control / choice	Purpose / action
Input limits	Up to 8 MB per file and 4096 x 4096 pixels; valid PNG/JPEG data.
Stored copies	Images fit within 64px, preserving aspect ratio. Maximum 128 imported icons per project.
Remove / unsupported	Removing a copy makes affected rows fall back to native icons. Use unpacked PNG originals for SVG, animation or unsupported rotated/tight atlas sprites.

Imported copies are stored in Editor metadata. The importer creates no new Assets, Resources or Addressables textures and no Player asset references.

04 - CHILD COLORS

12. Color a group deliberately

Child colors are explicit snapshots applied to existing objects. They are separate from applying the parent's complete style to children. Child-color edits preserve children's independent text and icon choices.



Control / choice	Purpose / action
Row only	Stops further child-color application. Previously painted child backgrounds remain.
Direct children	Colors the selected parent's immediate children only.
All descendants	Colors existing children at every depth beneath the selected parent.
Child color / Strength	Choose the fill and 0-1 opacity for the chosen scope. Edits apply the child-color action again.
Clear child colors	Removes backgrounds in the current child scope while preserving other child styling. Undo restores the action.

New and moved objects keep their style
Child tint is not continuously inherited from a parent. Reapply the action to include new children. Applying a reusable header can run its stored child-color recipe once; parent colors alone do not auto-style future descendants.

04 - COUNTS & DETAILS

13. Counts, tooltips, TODO and notes

Control / choice	Purpose / action
Count	The Text & roles button enables the direct child count.
All descendants	Counts all nested children. The selected root itself is never counted.
Active children	Counts immediate children that are activeInHierarchy.
Include inactive	Includes inactive objects in direct/descendant totals. The active-child total still counts active children only.
Tooltip	Adds an optional hover description. Long clipped names also have tooltip support.
TODO	Adds a metadata reminder to the row's presentation/tooltip information.
Notes	Stores project-local text in Editor metadata; no gameplay component is added.
Tree lines - global	Changes the same tree-line style used by Settings. It is not a per-row override.

Understand the count before using it

If a parent has two direct children and one of those has three children, the direct total is **2** and the descendant total is **5**. When the parent is inactive, its children can have activeSelf enabled while activeInHierarchy is false; the active-child total follows activeInHierarchy.

A useful note workflow

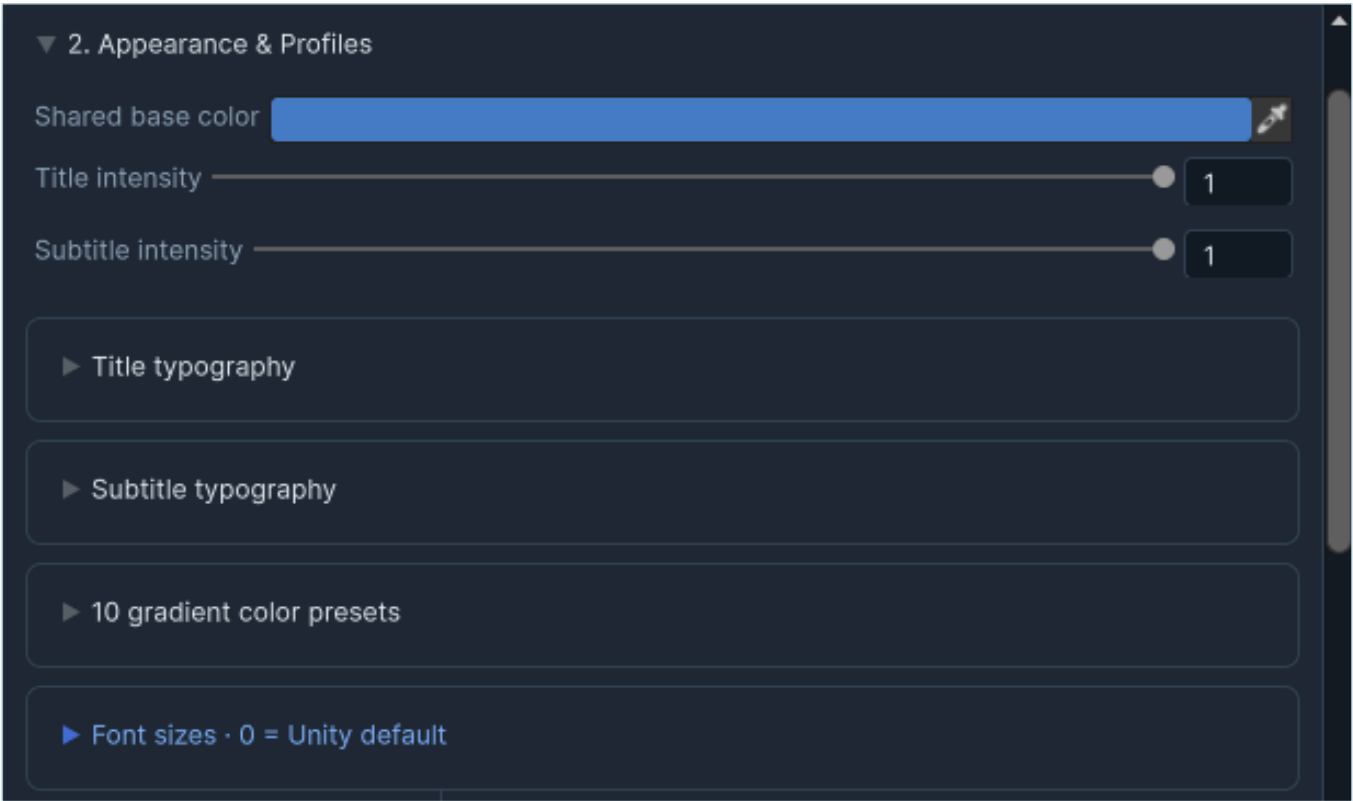
- 1 Open Advanced on the row and expand **Tooltip / TODO / Notes**.
- 2 Use Tooltip for a short purpose, such as "Lighting group for this scene". Use Notes for a longer reminder.
- 3 Enable TODO when a follow-up is needed. Save the scene and keep the metadata under version control with your team.
- 4 Hover the row to confirm the information. Do not expect notes to add another visible text line.

Counts can settle after changes

The package prepares row data and traverses counts through its cache work. Large hierarchy changes can require an update before displayed counts settle. Narrow rows can truncate indicators and names.

05 - APPEARANCE & PROFILES

14. Configure global role profiles



Global role profiles, typography, ten gradient presets and the font-size section.

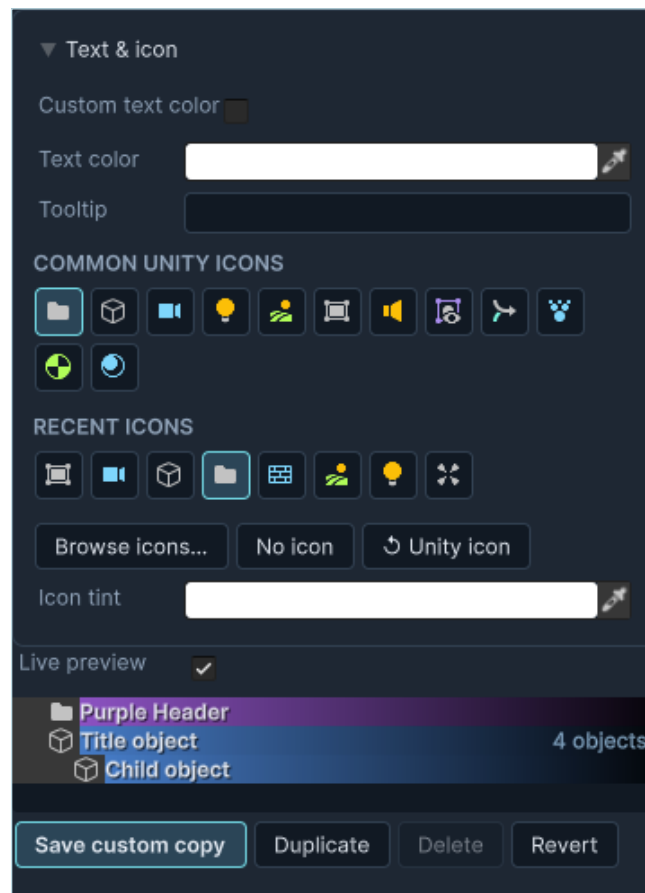
Control / choice	Purpose / action
Shared base color	A shared default color for Title and Subtitle profiles.
Title / Subtitle intensity	Separate strengths. Subtitle is constrained to no greater than Title intensity.
Title / Subtitle typography	Bold, Italic and alignment. Use selected reusable style copies the draft into that role profile.
10 gradient color presets	Configure the exact ten colors shown by the compact Quick Window.
Font sizes	Small, Normal, Emphasized: 0 uses Unity's default; positive sizes are clamped to 8-16.

Profiles are copied when applied

H/h in Quick apply profile snapshots. Changing Settings does not live-link or rewrite every previously styled row. Reapply the profile when you want existing assignments to adopt updated typography or intensity.

05 - REUSABLE STYLE EDITOR

15. Save your own header or subheader



The draft's text/icon choices, tooltip, live preview and explicit save actions.

- 1 Click **+ Header** or **+ Subheader** in Appearance & Profiles, or select a built-in style as your starting point.
- 2 Enter a unique Name. Set Type, Background, Strength, font choices, Child count and optional Accent border.
- 3 Choose child-color scope, text color, tooltip and icon. Inspect **Live preview**; its counts are illustrative.
- 4 Click **Save style** or **Save custom copy**. Open Advanced and apply the saved card from Headers or Subheaders.

Settings drafts need an explicit save

Row editing is immediate; reusable-style draft editing is not. Save before switching styles or closing Settings. Revert discards the current draft. Name changes rename the reusable style, not GameObjects.

05 - PRESET LIBRARY

16. Apply and maintain reusable styles

Factory styles are immutable definitions. They are available as Purple, Blue and Green headers and subheaders. Custom styles are stored locally to the project and can be exported as individual preset JSON files.

Control / choice	Purpose / action
Factory header	Title role, bold text, strong gradient and folder icon. It carries an All descendants child-color recipe at strength 0.18.
Factory subheader	Subtitle role, bold text, softer gradient at 0.32, accent border and folder icon.
Save custom copy	Creates a custom preset from a factory style without modifying the factory definition.
Save style	Writes a new custom style or updates the selected custom preset.
Duplicate	Creates a separately editable custom copy with an available name.
Delete	Removes a custom preset. The control is disabled for factory styles.
Revert	Reloads the selected saved style and discards unsaved draft changes.
Save style... in Advanced	Copies the current resolved row style into a new Settings draft. Name and save it there.

Apply a saved style

- 1 Select the intended object or objects and open Advanced.
- 2 Choose **Headers** or **Subheaders**, then click the style card. The complete style applies immediately, including a stored child-color action.
- 3 Make any row-specific adjustments afterward. Save your scene and verify the result at the hierarchy width you normally use.

Preset application is a snapshot

Changing or deleting a preset does not rewrite previously assigned rows. Reapply an updated card to refresh them. Named cards are in Advanced even when older Settings footer wording calls them the Quick Panel.

05 - SELECTION & GROUP ACTIONS

17. Edit several rows with clear scope

Multi-object editing is useful for consistent typography, icons and backgrounds. The clicked object is included; when it is part of the current selection, other selected editable GameObjects are included too. Check the panel heading and selection before a bulk command.

Control / choice	Purpose / action
Property edit	Changes the requested property on each selected row while preserving its other independent settings.
Profile / preset / paste	Applies a complete style snapshot to the targets. A template can also run its explicit child-color action.
Apply to children	Copies the first target's resolved complete style onto each selected parent's immediate children; excludes the parent rows.
Apply to descendants	Copies that complete style to all descendants of the selected parents; excludes the parent rows.
Reset selected rows	Clears the selected rows' own metadata assignments. Does not recursively clear their children.
Reset group and descendants	Clears selected roots and all descendant styles. Other unrelated scene objects are outside the scope.
Undo / Redo	Use Unity's Edit menu or normal shortcuts to reverse or repeat style commands.

Complete style or background only?

Need only a shared child background?

Use Child colors. Children retain their own icons, fonts and other presentation.

**Need every child to match the full parent style?**

Use Apply to children or Apply to descendants. The copied snapshot can replace their independent choices.

Nested selected groups

For child-color actions, outer groups are processed first. An explicitly selected nested group remains independent for its own descendants. Large commands can affect many rows; select and inspect the scope first.

06 - PREFAB STYLING

18. Share styles through prefab sources

Prefab styling is assigned to source object identities in Editor metadata. Instances can inherit that source presentation, while individual instance assignments take priority. Variants and nested prefab sources are resolved before their bases.

1. Own row assignment

An explicit scene-instance style wins. Editing an inherited row creates its own snapshot override.



2. Immediate prefab source, then base sources

If the row has no own style, source assignments are checked from the nearest source outward.



3. Optional global default

Used only when no own or prefab-source assignment resolves. Otherwise Unity's native appearance remains.

- 1 Customize a prefab instance in Advanced until its row looks correct.
- 2 Click **Style prefab source**. The corresponding source object receives that resolved style.
- 3 Use **Reset instance override** on the edited instance so it returns to source inheritance.
- 4 Check another existing instance and a newly created instance of the same prefab. Use a variant source when its presentation should differ from the base prefab.

Identity matters

Moving or renaming an asset while retaining its GUID preserves references. Rebuilding a prefab object with a new local ID may require reassignment. A duplicated prefab asset has a new GUID and independent assignments. A local instance override keeps its chosen style until reset.

06 - HIERARCHY FEATURES

19. Tree lines, activation and defaults

Control / choice	Purpose / action
Activation control	Adds a small active-state control to the row. Clicking it toggles <code>GameObject.activeSelf</code> , records Undo and prefab instance modifications.
Tree lines	Classic, Minimal, Dotted or None. These are conservative local connectors in the callback gutter.
Line color / opacity	Choose a shared line color and 0-1 opacity. Advanced's Tree lines style uses the same global setting.
Collapse All	Requests native Alt + Left keyboard collapse on loaded scene roots, then restores selection/focus.
Global default style	Allows fallback styling for objects with no own assignment or prefab source style.
Use current profile as default	Copies the selected Settings draft into the global default snapshot.
Right-side space	Data & Maintenance reserves 0-96 extra points for another extension's indicators. Unity native indicators are reserved automatically.

Choose a readable layout

- 1 Start with your normal hierarchy width and set Tree lines to the style you prefer.
- 2 Enable activation only if row-level active-state editing helps your workflow. Inspect active and inactive rows.
- 3 Choose a subtle fallback style if using Global default style. Explicit and prefab-source styles take priority.
- 4 If indicators crowd names, widen Hierarchy, disable optional row features or increase extra right-side space for another extension.

Public Editor API limits

Search results, narrow gutters and horizontal scrolling can omit local connectors. Collapse All cannot guarantee scene-header or hidden-descendant expansion state through public APIs. Clear search and use Unity's native foldout gestures when necessary.

The activation indicator reflects `activeSelf`. An object with `activeSelf` enabled can still be `inactiveInHierarchy` under an inactive parent. Activation edits are object changes; save the scene or prefab as appropriate.

06 - COMPONENT MINIMAP

20. Inspect components from a row

Enable **Settings > Components & Icons > Component Minimap** to display component icons beside rows. The minimap is optional and uses the available right-side width.

- 1
- Hover a component icon to identify it. Rows show up to four slots when enough width is available.
- 2
- Alt + left click a component icon** to open Component Mini Editor. This component gesture is Alt-click even if the row-opening gesture uses another modifier.
- 3
- If the last visible slot represents overflow, Alt-click it and choose a component from the menu.
- 4
- Edit the component's visible serialized properties. Unity Undo and prefab override recording follow serialized property editing.
- 5
- Use **Open standard Inspector** for managed references, custom Inspector-only controls or components that require their normal Inspector.
- 6
- Save the affected scene or prefab after changing component data.

Control / choice	Purpose / action
Visible properties	The mini editor presents serialized fields and skips the script reference field.
Managed references	Use the standard Inspector fallback; managed-reference properties are skipped in the mini editor.
Custom Inspector behavior	Custom buttons, bespoke editors and Inspector-only logic may require the full Inspector.
Narrow hierarchy	The available width limits visible component slots. Overflow provides access to the remaining components.
Deleted target	The mini editor closes when its component is no longer available.

Component changes affect your project

This utility edits actual component values, unlike metadata-only styling. Check the target component and save changes just as you would when using the normal Inspector.

06 - SCENE NAVIGATION

21. Set up a compact scene bookmark bar

- 1
- Open **Settings > Scene Navigation** and turn on **Enable scene navigation**.
- 2
- Click **Open compact scene bar**. Dock the Scene Navigation window beside or above Hierarchy, wherever it fits your layout.
- 3
- Assign a saved SceneAsset to **Bookmark scene**. Alternatively, open a saved scene and use the bar's **+** button to bookmark it.
- 4
- Click a bookmark to switch scenes. Unity asks about modified scenes before opening the chosen scene in **Single** mode.
- 5
- Right-click a bookmark to **Move left**, **Move right** or **Remove bookmark**.
- 6
- Use **Recent scenes** to open a recently opened/saved scene, or the collapse control to collapse loaded scene roots.

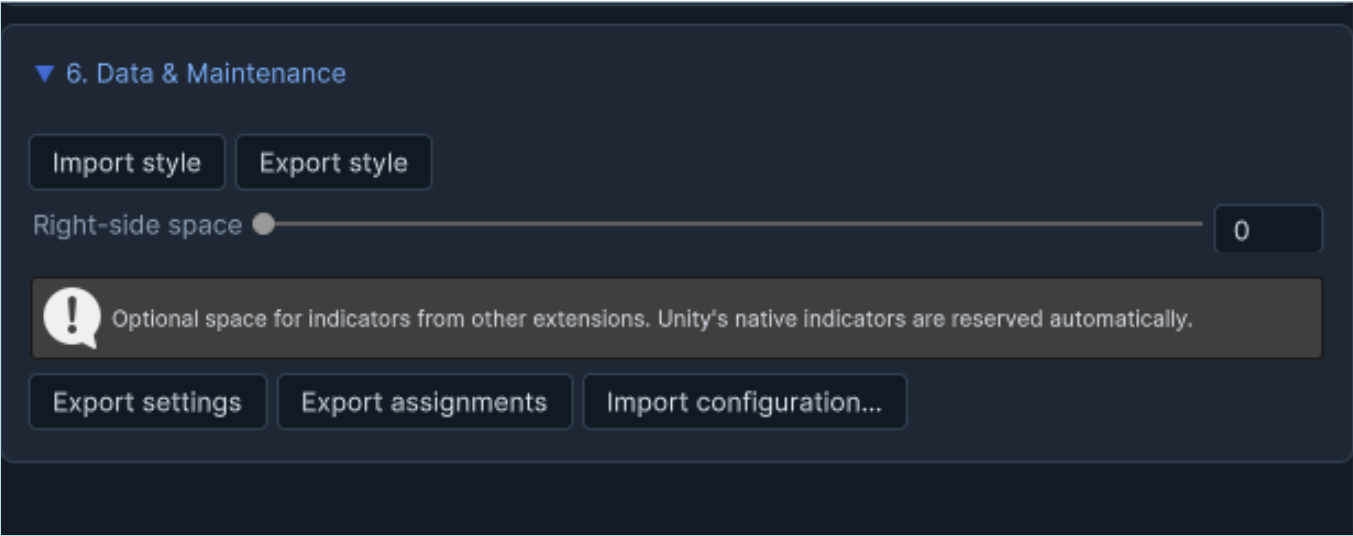
Control / choice	Purpose / action
Bookmark storage	Scene GUIDs keep references stable across ordinary moves and renames with matching .meta files.
Active scene	The current active scene bookmark is highlighted.
Missing scene	An unresolved bookmark is shown as Missing scene and can still be removed by right-click.
Recent list	Keeps up to eight saved scene references. A scene without a saved path cannot be bookmarked by +.
Limits and Play Mode	Up to 256 bookmarks. Scene switching is ignored during Play Mode or a Play Mode transition.

A separate dockable utility

The bar is an auxiliary Editor window. Public Unity APIs do not expose safe native Hierarchy toolbar insertion. Scene switching uses normal unsaved-scene confirmation and opens a single scene, so save before switching.

07 - DATA & MAINTENANCE

22. Choose the right export



Settings > Data & Maintenance. Single-style and full configuration exports have different scopes.

Control / choice	Purpose / action
Export style	One selected reusable style/draft in preset JSON. Use for a single portable recipe.
Import style	Imports preset JSON as a new custom style with an available name; does not overwrite an existing factory/custom identity.
Export settings	Global options, Title/Subtitle profiles, ten Quick colors, imported icon payloads, bookmarks and custom presets.
Export assignments	Project object and prefab style records only. Their references are project-specific.
Import configuration...	Reads a settings or assignments file, validates it, creates backups, then offers Merge or Replace.

Back up both categories

For a complete project backup, export settings and assignments as separate files. Individual preset files do not include the project icon library; include settings when a style depends on private imported icons.

07 - IMPORT & RECOVERY

23. Merge, replace and restore safely

Configuration files identify their category and version. Validation occurs before mutation. The importer records Undo and captures pre-import state; failed imports restore snapshots. Unresolved references are reported and retained for recovery.

Control / choice	Purpose / action
Merge settings	Incoming option/profile values apply. Bookmark and imported-icon collections combine; matching preset IDs update and unrelated custom presets remain.
Replace settings	Replaces global settings and custom presets from the selected file. Object assignments remain a separate category.
Merge assignments	Updates matching object IDs and adds incoming assignments while retaining other assignment records.
Replace assignments	Replaces object assignment records from the selected file. Global settings/custom presets are outside this category.
Automatic backup	Before an import, both categories are written under ProjectSettings/ERAGANX.HOP.Backups/ with timestamped JSON filenames.
Icon ID conflict	Merge rejects the same imported-icon ID with different image payloads. Use an intentional Replace or import images separately.
Missing references	Reports unresolved scene, asset or icon references. Matching GUIDs/local IDs are required for assignments to resolve.

- 1 Export current settings and assignments, then choose **Import configuration...** and the desired JSON file.
- 2 Read the Merge/Replace dialog and choose the operation appropriate for that category. Merge still applies incoming global option values.
- 3 Inspect the status and any missing references. Check styled rows, imported icons, presets and bookmarks.
- 4 If the result is unsuitable, use Undo or import the matching pre-import backup. Keep a copy of your current data before manual file recovery.

Single preset JSON is limited to 64 KB. Configuration input is limited to 24 MB, with up to 2,000 custom presets or 100,000 assignment records, depending on the file category.

07 - PERSISTENCE & TEAMS

24. Keep styles across saves and updates

Control / choice	Purpose / action
Style database	ProjectSettings/ERAGANX.HierarchyOrganizerPro.asset stores assignments and custom presets.
Options database	ProjectSettings/ERAGANX.HierarchyOrganizerPro.Options.asset stores global options, profiles, icons and scene references.
Database backup	The style database has a .bak recovery copy. Configuration imports also create JSON backups in ERAGANX.HOP.Backups.
Saved scenes / prefabs	Persistent references use asset GUIDs and local object IDs. Preserve the assets' .meta files.
Unsaved scenes	Assignments are temporary until saved identities become available. Unsaved objects are not durable across project closure.
Moving and Save As	Live objects can promote/migrate styles as identities change. Unresolved records are retained instead of automatically pruned.
Team source control	Keep the relevant ProjectSettings files, scene/prefab assets and their .meta files together. Resolve conflicting metadata deliberately.
Package updates	Back up both configuration categories first. Factory definitions do not overwrite user custom presets.

Transfer to another project

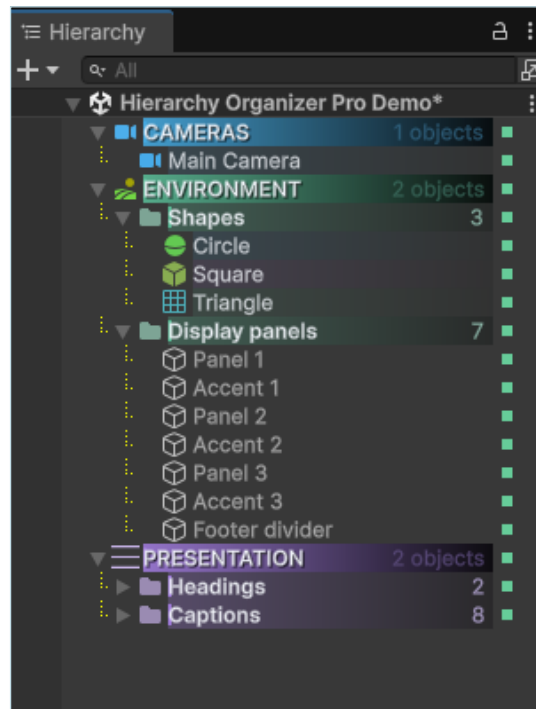
- 1 Import the Editor package into the destination project. Export/import Settings for reusable styles and their icon payloads.
- 2 Transfer scene/prefab assets with their matching GUIDs and local object identities before importing Assignments.
- 3 Check the missing-reference report. Recreated objects with new identities need new assignments; a similar object name alone is not a match.
- 4 Open the scenes, inspect the rows and save the project data.

Disable and remove deliberately

Enable Styling can hide the styling display without deleting records. Before removing the extension, keep configuration exports or metadata copies if you plan to restore it. Removal and updates should be checked in your own project.

08 - COMPLETE SCENE WORKFLOW

25. Recreate the example organization



The example uses clear group colors and actual object names. Styling alone does not create this hierarchy structure.

- 1 Use your existing groups or create/parent them with normal Unity commands. The illustrated scene has CAMERAS, ENVIRONMENT and PRESENTATION roots.
- 2 Apply blue, green and purple Title styles to major groups. Pick icons that communicate their purpose.
- 3 Use softer Subtitles for Shapes, Display panels, Headings and Captions. Apply child colors explicitly where they improve scanning.
- 4 Add direct child counts to major groups and relevant subgroups. Inspect names, selection and inactive states at your normal hierarchy width.
- 5 Save reusable variations for your project, save the scene, and export settings plus assignments before sharing the presentation.

Demo assets and assignments are separate

The inspected source project has a Demo scene, but publisher demo/export contents can differ. Copying a scene alone does not transfer ProjectSettings styling records. The supplied screenshots show Editor presentation; a Player displays only the scene's actual geometry and components.

08 - EDITOR C# EXAMPLE

26. Automate a selected-row style

Optional for tool developers: place this script under your own **Editor** folder. It uses the package's public Editor APIs and Unity selection. If your Editor asmdef is not automatically referencing the package, add **ERAGANX.HierarchyOrganizerPro.Editor**.

```
#if UNITY_EDITOR
using UnityEditor;
using UnityEngine;
using ERAGANX.HierarchyOrganizerPro.Core;

public static class ProjectHierarchyStyle
{
    [MenuItem("Tools/Project/Style Selected Group")]
    private static void ApplyGroup()
    {
        var targets = Selection.gameObjects;
        var style = new HierarchyStyle
        {
            role = StyleRole.Title,
            kind = RowKind.Section,
            bold = true,
            backgroundEnabled = true,
            background = new Color(0.27f, 0.48f, 0.77f),
            opacity = 1f,
            directCount = true,
            childColorScope = ChildColorScope.Descendants,
            childBackground = new Color(0.27f, 0.48f, 0.77f),
            childOpacity = 0.18f
        };
        StyleCommands.ApplyTemplate(targets, style,
            "Style selected project groups");
    }

    [MenuItem("Tools/Project/Reset Selected Row Styles")]
    private static void ResetRows()
    {
        StyleCommands.Apply(Selection.gameObjects, null,
            ApplyScope.Selection, "Reset selected styles");
    }
}
#endif
```

Editor code, not gameplay code

These APIs are in an Editor-only assembly. Keep references out of runtime assemblies and Player code. Commands filter non-editable objects and refuse row editing during Play Mode. Save scene identities and preserve the metadata after authoring.

This example matches inspected method signatures; it is not a runtime feature. A compilation or visual test of a newly added integration remains your project's responsibility.

08 - EDITOR API REFERENCE

27. Public commands at a glance

Namespace: **ERAGANX.HierarchyOrganizerPro.Core**. Assembly: **ERAGANX.HierarchyOrganizerPro.Editor**.
Prefer property edits when preserving independent row choices matters.

Control / choice	Purpose / action
StyleResolver.Resolve(go)	Returns the resolved own/source/default style, or null. Clone the returned style before direct customization.
StyleCommands.Modify(...)	Modify(GameObject[], Action<HierarchyStyle>, ApplyScope = Selection, string action). Edits chosen properties and preserves others.
StyleCommands.Apply(...)	Apply(GameObject[], HierarchyStyle, ApplyScope, string action). Copies a complete style to the scope. null resets own assignments.
StyleCommands.ApplyTemplate(...)	ApplyTemplate(GameObject[], HierarchyStyle, string action). Applies the template and its explicit child-color recipe.
StyleCommands.ModifyGroup(...)	ModifyGroup(GameObject[], Action<HierarchyStyle>, bool applyChildColors, string action). Edits properties and optionally paints children.
ClearChildColors(selected)	Clears backgrounds in each selected group's remembered child-color scope.
Copy(go) / ReadClipboard()	Writes/reads the package's clipboard payload. ReadClipboard returns null for invalid content.
ApplyPrefabSource(go)	Assigns the resolved style to the corresponding prefab source. Customize the object first.
PresetLibrary	Factory, All, Find; SaveCustom, SaveCopy, Update, Rename, Delete; Export, Parse and ImportFile.
HierarchyStyle.Clone / Normalize	Clone before editing shared data. Normalize validates enums/ranges and derives the horizontal gradient.

```
// Keep each selected object's other style choices.
StyleCommands.Modify(Selection.gameObjects,
    s => { s.bold = true; s.uppercase = true; },
    ApplyScope.Selection, "Emphasize selected rows");
```

ApplyScope: Selection = targets only; DirectChildren = immediate children; Descendants = all children excluding roots; Subtree = roots plus all descendants. ChildColorScope has None, DirectChildren and Descendants.

09 - TROUBLESHOOTING

Fix interaction and styling issues

Control / choice	Purpose / action
Tools menu is missing	Wait for import/compilation. Resolve Console errors and confirm the Editor assembly and UI resources are present under the package root.
Nothing is styled	Enable Styling in Tools or Settings. Work in Edit Mode, choose an editable object and assign a color or profile.
Alt-click opens the wrong window	Settings > Window & Interaction chooses Quick or Advanced. The Quick arrow also opens Advanced for the current targets.
Alt-click does nothing / conflicts	Use the object row rather than the foldout arrow. Choose Alt Shift Click or Control Shift Click if another extension consumes the default gesture.
Several rows changed	The clicked object was part of the current selection. Select one object alone for a single-row edit.
The popup closes	Outside focus finishes editing. Keep pointer input inside the Advanced owned color picker; use Done/Escape to return.
Subtitle has no second line	Subtitle is a role at Unity's native row height. Descriptions appear in tooltips, not a second text line.
New children have no tint	Child coloring is an explicit snapshot. Reapply the child-color action to include newly added or moved objects.
Row only did not erase old tints	Row only stops future child-color application. Use Clear child colors in the previous scope, or reset the intended rows.
Reset still shows styling	Reset removes the own assignment. A prefab-source style or enabled global default may still resolve.
Preset edits did not update rows	Assignments are snapshots. Save the changed preset, then reapply its card to the intended objects.

A useful isolation check

Select one saved, ordinary GameObject, clear the Hierarchy search, enable styling and apply one visible Quick color. Then add profiles, icons and optional features one at a time. Undo lets you restore your prior presentation.

09 - TROUBLESHOOTING

Fix data, icons and crowded layouts

Control / choice	Purpose / action
Styles disappear after reopening	Save unsaved scenes. Preserve both ProjectSettings metadata files. Keep a current copy before restoring the database .bak or importing a backup.
A saved style is missing	Save the Settings draft explicitly. Check Headers versus Subheaders in Advanced. Inspect any save error in the footer.
Prefab style does not propagate	Use Style prefab source, then Reset instance override. Existing local instance assignments take priority over source styles.
Style lost after rebuilding a prefab	A new local object ID may no longer match the old assignment. Reassign the style to the new source object.
Custom icons are absent	Enable Icon package. Check whether the imported copy still exists. Missing images fall back to the native icon.
Icon import fails	Use a valid PNG/JPEG within 8 MB and 4096px input limits. Stay within 128 stored copies. Use unpacked originals for unsupported sprites.
Preset imported but icon is missing	A single preset does not carry private image payloads. Import the corresponding Settings configuration or import/reassign the image.
Imported assignments show missing assets	Transfer matching scene/prefab GUIDs and local IDs. Object names are not identity keys. Check the import status.
Names and counts are crowded	Widen Hierarchy, reduce optional component/activation features, or reserve right-side space for another extension.
Tree lines or collapse differ in search	Clear search. Public APIs provide local row geometry, not all visible-tree or expansion details. Use native foldouts where necessary.
A mini-editor control is missing	Use Open standard Inspector for managed references and custom Inspector-only behavior.

For a reproducible issue, include the exact Unity version, OS, Editor theme/scaling, hierarchy width, other hierarchy extensions, steps, Console errors and a small saved scene/prefab example.

10 - COMPLETION & FAQ

Your first-project checklist

- 1

The package compiles; the Tools menu is visible and Enable Styling is on.
- 2

The opening gesture and Quick/Advanced choice fit your Editor setup.
- 3

A Title, a Subtitle and an ordinary row are readable with selected and inactive states.
- 4

Your chosen icons, counts and child-color scope work at your normal hierarchy width.
- 5

Custom reusable styles are explicitly saved and available in Advanced.
- 6

Prefab source inheritance and instance reset work for your intended prefabs.
- 7

Any component/activation edits are reviewed and the affected scenes/prefabs are saved.
- 8

Scene bookmarks, settings and assignments have been checked; both export categories are backed up.
- 9

Reopen the project and confirm persistence. Verify other Unity versions, themes, scales and extensions in your own environment.

Question	Answer
Does it change object names or parenting?	Styling keeps actual GameObject names and hierarchy structure. Make structural changes with normal Unity tools.
Do colors appear in my game?	Hierarchy presentation is Editor-only. Activation and component edits affect actual objects.
Can I use it with 2D and 3D scenes?	It styles GameObject hierarchy rows; styling has no render-pipeline setup requirement.
Are presets portable?	Style recipes/settings are portable. Object assignments need matching identities; private icons need their Settings payload.
Is it automatic grouping?	You apply styles manually. Child coloring is a snapshot; prefab-source inheritance is a separate workflow.

10 - GLOSSARY & SOURCE SCOPE

Terms, versions and getting help

Term	Meaning
Title / Subtitle	Presentation roles for real names at Unity's native row height.
Profile / reusable style	Global role defaults copied on application / a saved named style recipe.
Draft / assignment	Unsaved Settings edits / metadata applied to a specific object identity.
Child snapshot / inheritance	An explicit fill action on existing children / resolution through a prefab source.
Own override / fallback	A row's explicit snapshot / the next prefab source or optional global default.
GUID / local ID	Asset identity / object identity inside that asset. Names alone do not identify assignments.
Settings / assignments export	Portable options, profiles, icon payloads and presets / project object/prefab records.

Guide source and version scope

Prepared in English on **13 October 2026** from the inspected package at **Assets/ERAGANX/Hierarchy Organizer Pro/**, its documentation and thirteen supplied screenshots. The Water Reflection guide supplied by ERAGANX informed the layout and reading structure; its product-specific instructions do not apply to this tool.

Source target: **Unity 6000.3.13f1**. Checked sources include Settings/Quick/Advanced UI, style commands/resolution, preset library, icon importer/browser, component editor, scene navigation and configuration transfer. Screenshots can contain example values and wording retained from earlier UI revisions.

Compatibility and performance scope

Other Unity versions require their own verification. Native-height rows, local tree connectors and collapse behavior follow the public Editor API limits described here. This guide does not promise lag-free large scenes, zero overhead or untested platform behavior; evaluate responsiveness in your own project.

Getting help

Use the support details provided by ERAGANX on the product or publisher listing. Include a small reproduction, exact Unity version, relevant settings, screenshots and Console messages. The bundled Documentation folder includes further troubleshooting and technical notes.

Google Material Icons are included under Apache License 2.0. See the package's Licenses/Third-Party Notices.txt and GOOGLE_MATERIAL_ICONS_LICENSE.txt.