Discovery Style Guide

Ansys, Inc.
2600 ANSYS Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
http://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 2023 R2
July 2023

Ansys, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001:2015 companies

Copyright and Trademark Information

Disclaimer Notice

U.S. Government Rights

Third-Party Software

Published in the U.S.A.

Contents

Introduction

This document provides guidelines for:

- Colors
- Themes
- Typography
- Icons

See the separate Quick Reference for Tool Usage for guidance on tool workflows and behavior.

Colors

Color Palettes and Their Usage

Details for all color palettes (including Hex and RGB info) are included in a separate Excel file (Ansys Design Language Color Palettes). Guidelines for using the palettes are as follows:

Application	**Palette Usage**
UI elements such as • buttons • panel backgrounds • indicators of status and state NOT icons or cursors	Main UI Control Colors per Theme All UI Control Colors • The "Main UI Control Colors per Theme" palette includes the main colors that are used in each theme and how they are used (see also Figures 1-4 below). • If you need more colors, choose from the "All UI Control Colors" palette. • For guidance on colors with specific meanings, see the semantic colors in Figure 5 below.
Icons and cursors	Theme-Dependent Icon Colors Additional Colors • Use the "Theme-Dependent Icon Colors" palette for icon colors that should change with the theme, as described in Theme-Dependent Coloring • If you need more colors, choose them from the "Additional Colors" palette.
Predefined colors the user can assign	User-Assignable Colors Additional Colors • Use the "User-Assignable Colors" palette as the predefined colors in the color picker that appears in a product when a user has the option to choose a color explicitly (e.g., for a chart line). See Assignable Colors for details. ○ There is a vivid palette and a muted palette. • If you need to include more colors for the user to choose from, use colors from the "Additional Colors" palette.

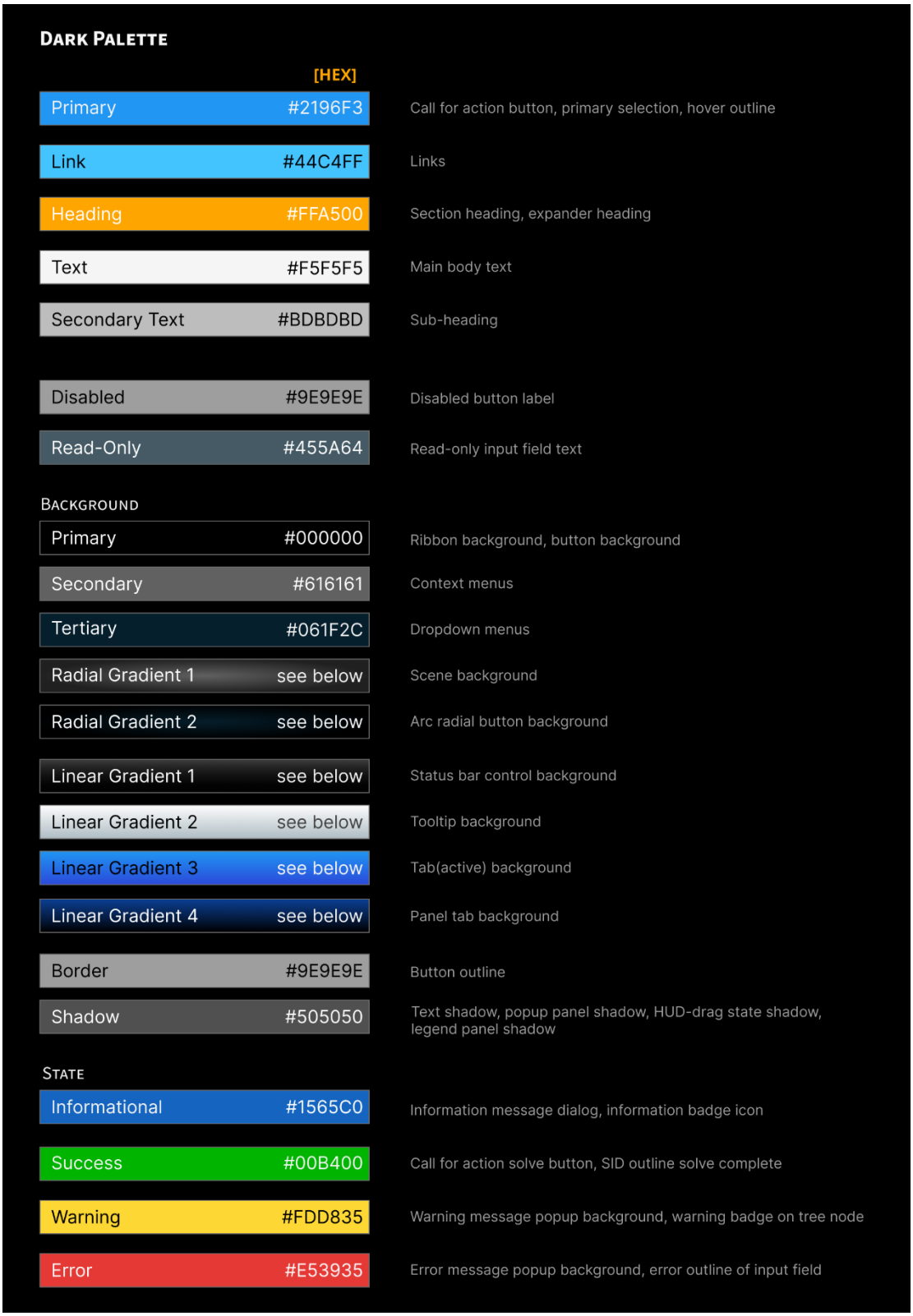


Figure 1: Main Colors Used in the Dark Theme

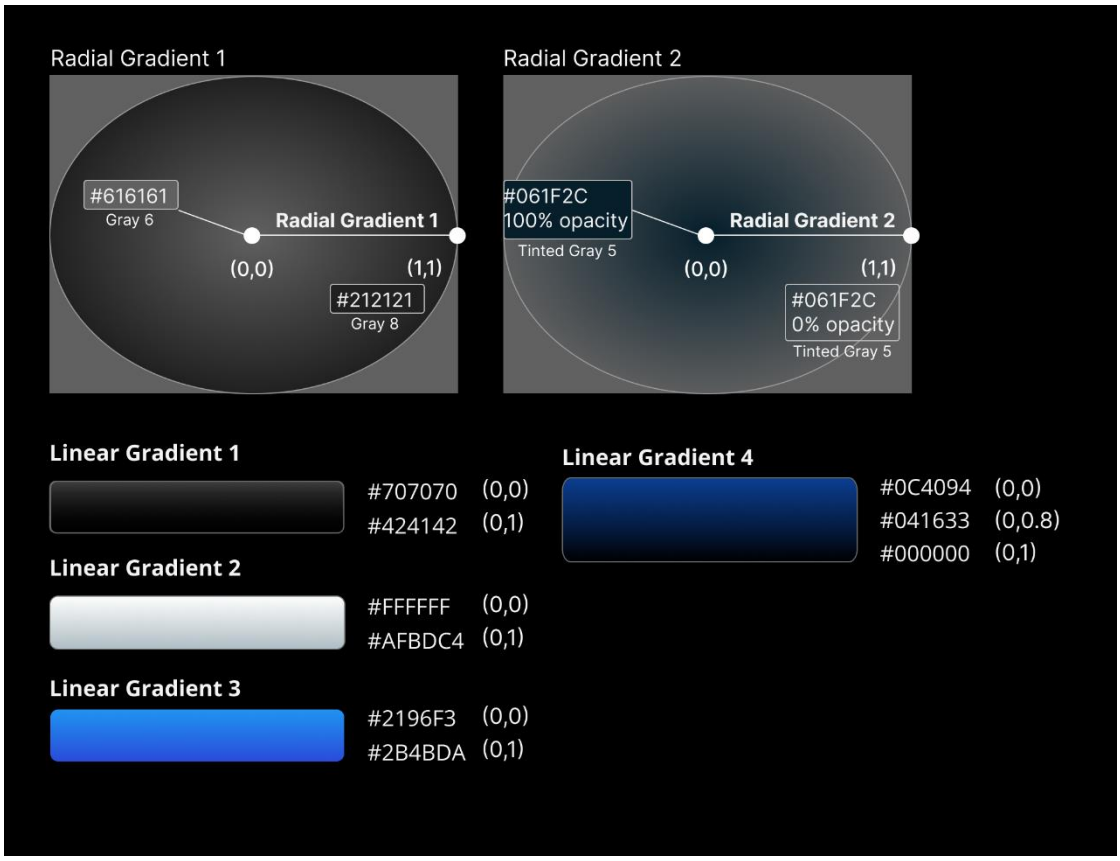


Figure 2: Main Color Gradients Used in the Dark Theme

Light Palette

	[HEX]	
Primary	#2196F3	Call for action button, primary selection, hover outline
Link	#2B4BDA	Links
Heading	#F3581C	Section heading, expander heading
Text	#212121	Main body text
Secondary Text	#505050	Sub-heading
Disabled	#727272	Disabled button label
Read-Only	#B3BEC7	Read-only input field text

Background

Primary	#FFFFFF	Button background
Secondary	#F5F5F5	Context menus, Dropdown menus
Tertiary	#CFD8DC	Ribbon background
Radial Gradient 1	see below	Scene background
Radial Gradient 2	see below	Arc radial button background
Linear Gradient 1	see below	Status bar control background
Linear Gradient 2	see below	Tooltip background
Linear Gradient 3	see below	Tab(active) background
Linear Gradient 4	see below	Panel tab background
Border	#727272	Button outline
Shadow	#BDBDBD	Text shadow, popup panel shadow, HUD-drag state shadow, legend panel shadow

State

Informational	#1565C0	Information message dialog, information badge icon
Success	#00B400	Call for action solve button, SID outline solve complete
Warning	#FDD835	Warning message popup background, warning badge on tree node
Error	#E53935	Error message popup background, error outline of input field

Figure 3: Main Colors Used in the Light Theme

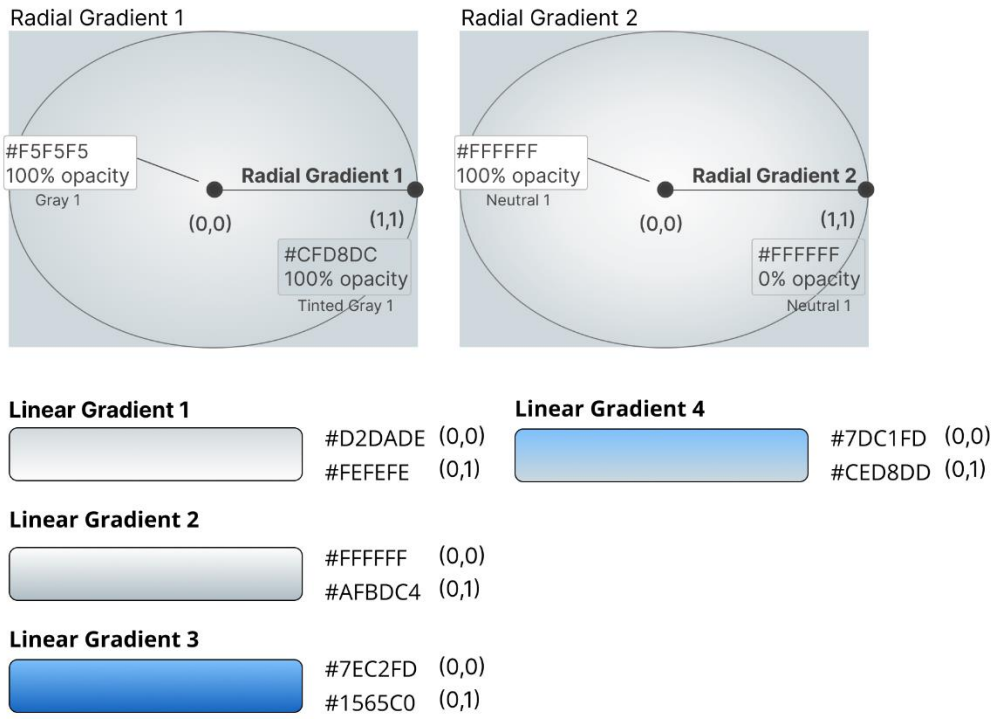


Figure 4: Main Color Gradients Used in the Light Theme

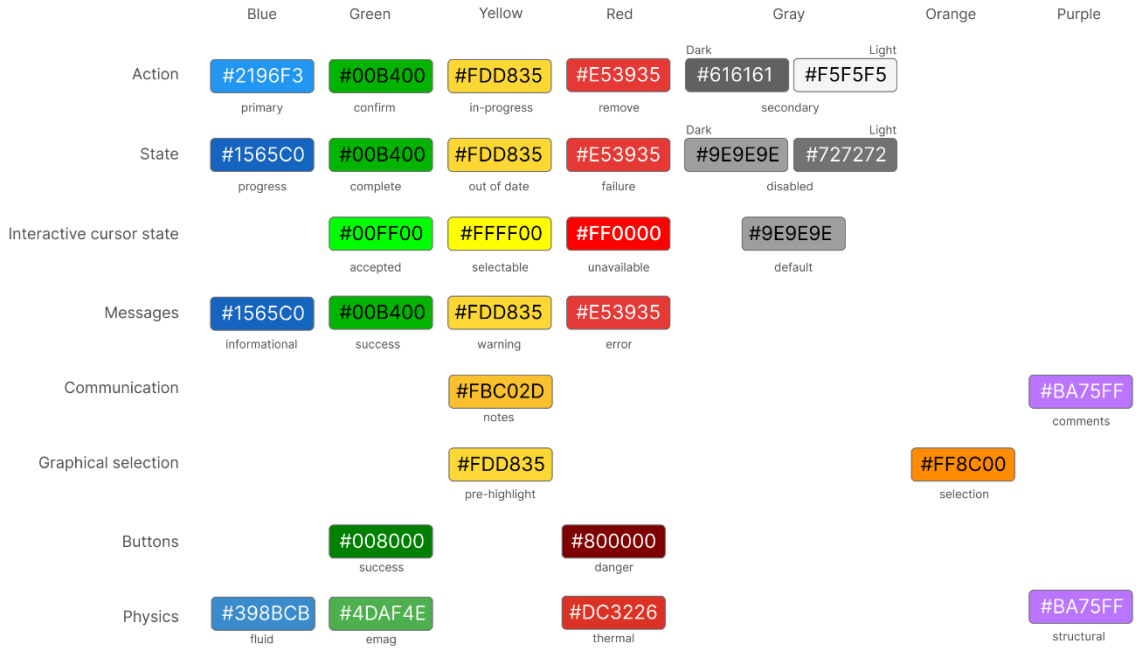


Figure 5: Semantic Colors Used in All Themes

Figure 6: Example of Color Usage in a Product UI (Dark Theme)

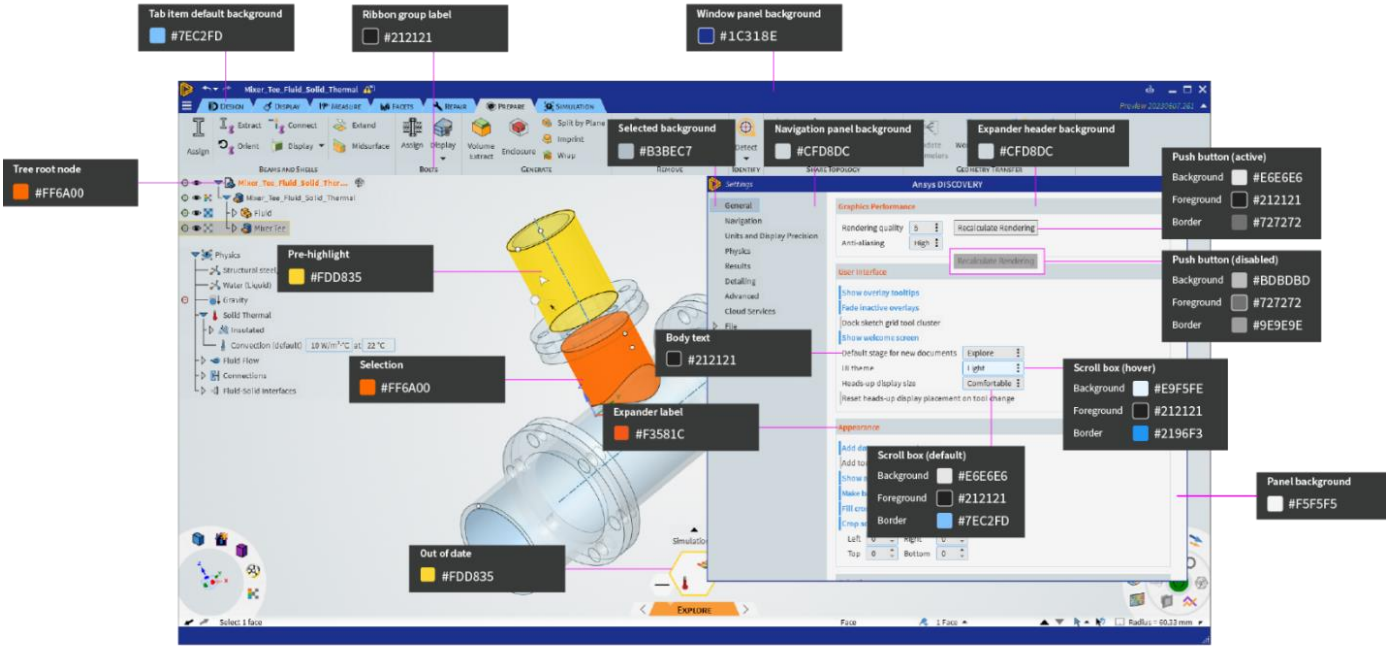


Figure 7: Example of Color Usage in a Product UI (Light Theme)

Icon and Cursor Colors

Icon Colors

Some of the colors used for icons are constant across themes, while others change between themes to better match the altered palette. Certain colors used in the icons have assigned meanings, which are discussed in detail in the Icons section. See Color Palettes and Their Usage for details about which colors are to be used in icons.

Figure 8:Shades of Green and White as Examples of Icon Colors.

Cursor Colors

The cursor must always be visible, no matter the color of the UI element below it. Unlike stationary icons, cursors move about the screen and are designed to maintain good contrast against any possible background.

In most cases, the cursor is white with a black outline. The white stands out strongly on dark backgrounds, while the black outline ensures that the cursor will remain visible in a light setting. In certain situations where there's a need to distinguish between a pair of tools that are similar in functionality and use the same shape cursor (e.g., Pull and Move tools for geometry modeling), you can reverse the cursor color treatment, so the fill color is black, and the outline is white. This change to the cursor reminds the user of the tool they are using as they make their selection.

Cursors can also have a thin outline of a static color that maintains the cursor's shape against varying backgrounds or UI elements. The outline alone might not be visible against backgrounds of similar color, but the infill of a contrasting color makes the cursor visible.

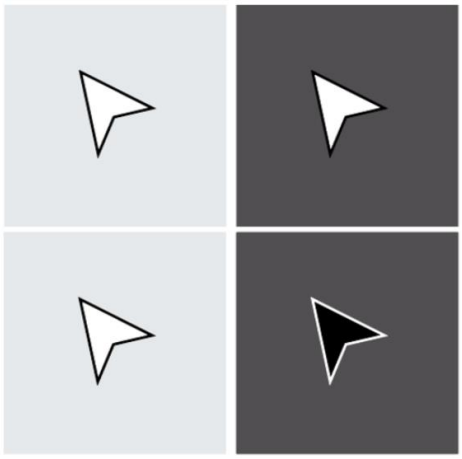


Figure 9: Cursor Outline Ensures Visibility on Different Backgrounds

Theme-Dependent Coloring (coming soon)

An icon may need to use different colors based on the selected theme. To ensure that the icons are rendered appropriately in all themes, the designer must create a separate version of the icon for each theme. Based on the active theme, the appropriate version of the icon is used.

The following example shows how the light and dark versions of an icon would look:

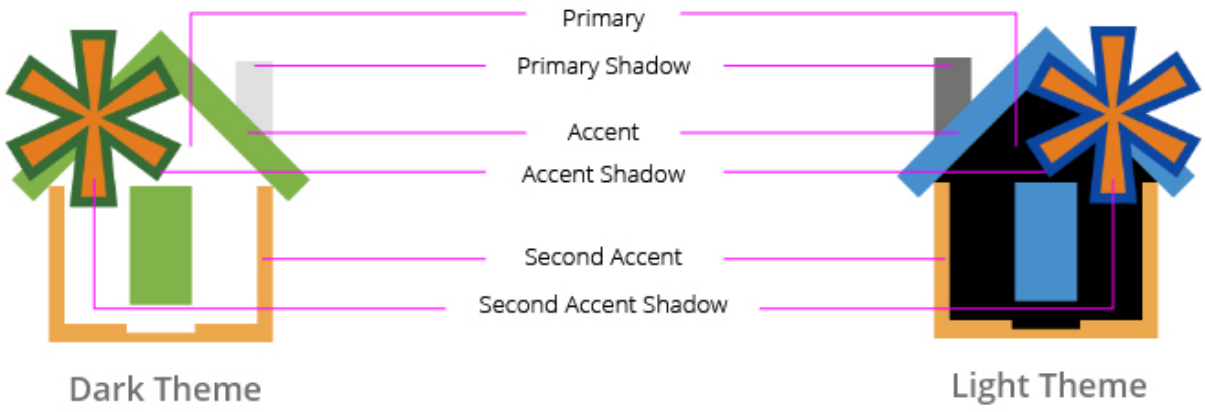


Figure 10: Example of Theme-Dependent Coloring in an Icon

The theme-dependent colors for icons are shown below.

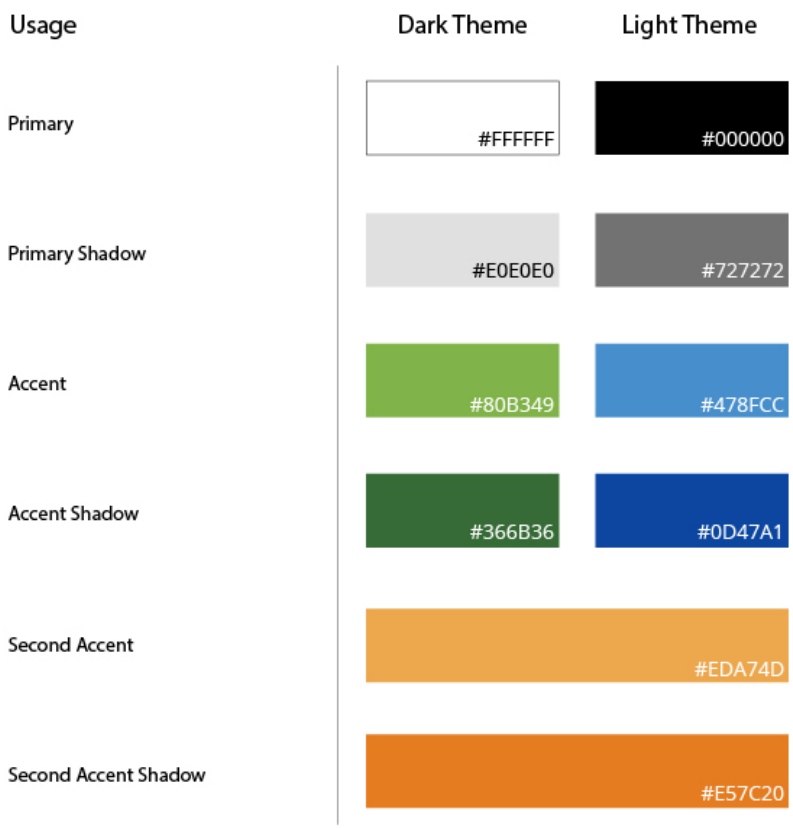


Figure 11: Theme-Dependent Colors for Icons

Status Colors

These are colors that have been assigned meanings corresponding to the type of information being conveyed in a message or other UI element. As specified in the Main UI Control Colors per Theme palette described in Color Palettes and Their Usage:

- blue for information
- green for good
- red for errors
- yellow for warnings

Note that these colors reinforce a message, but they should not be the only cue conveying information since a portion of the population has some form of color vision deficiency.

Figure 12: Red for Errors, Yellow for Warnings, Blue for Information

Background Colors

Background colors vary by theme (see Color Palettes and Their Usage), and they may include transparency or gradient effects. They cover most of the UI and strongly influence the rest of the palette, as other elements must adjust to maintain proper contrast against the theme's base colors.

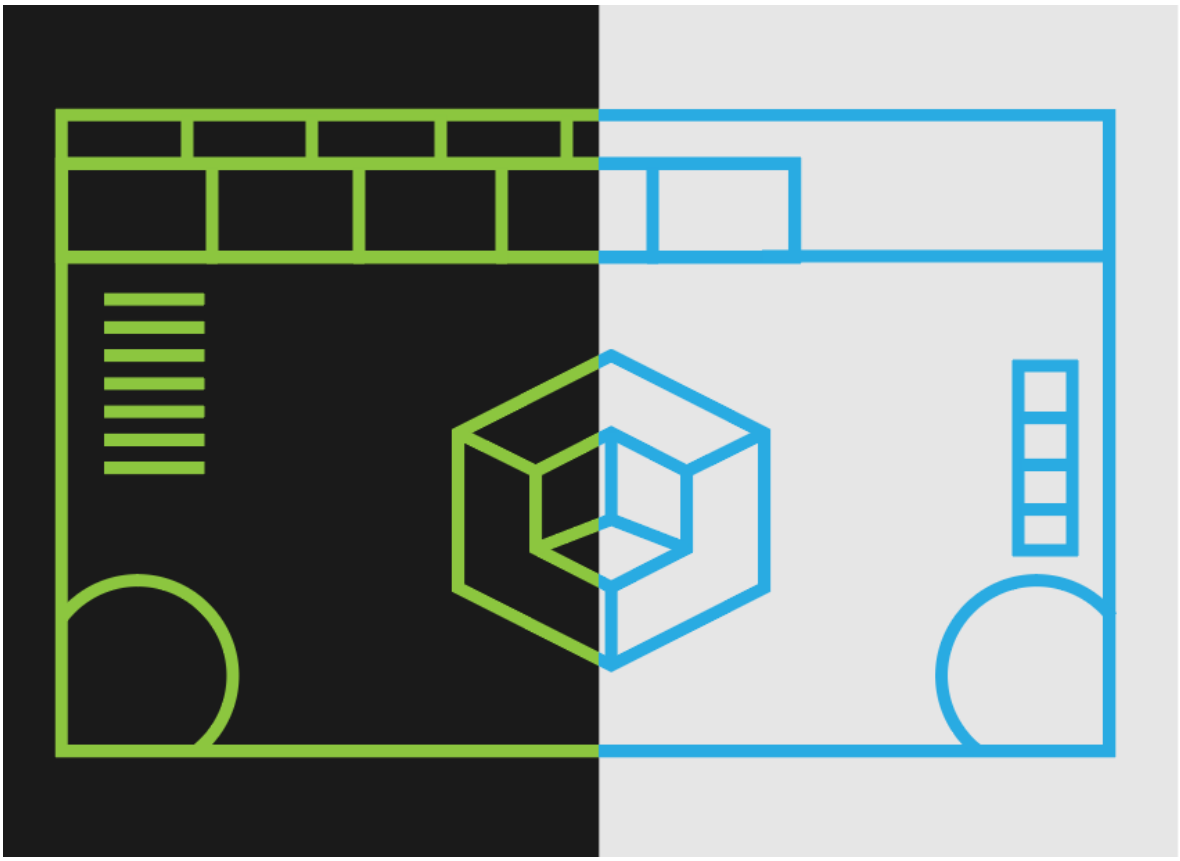


Figure 13: Background Colors are Theme-Dependent

Assignable Colors

In most products, there are situations where the user can choose colors as part of their workflow. Examples include the colors of chart lines representing different data series and the colors of the bodies of a 3D geometric model. There are two color palettes offered for these situations: the user can choose from a vivid color palette or a muted color palette. See Color Palettes and Their Usage for details.

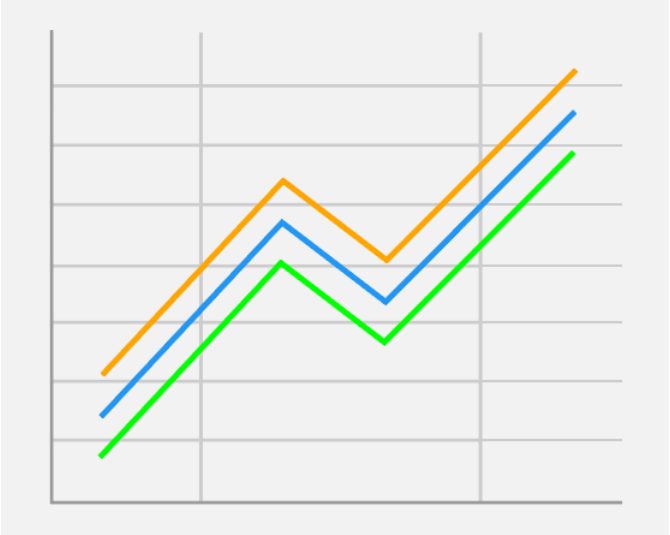


Figure 14: Chart Lines with Colors Assigned by the User

Note that 3D geometry models also have colors assigned to their bodies based on their defined materials; the user can choose whether to view the material-based colors or the user-specified colors. In either case, colors on the 3D model change to indicate what the user has selected, as indicated in the Graphical Selection row of Figure 5.

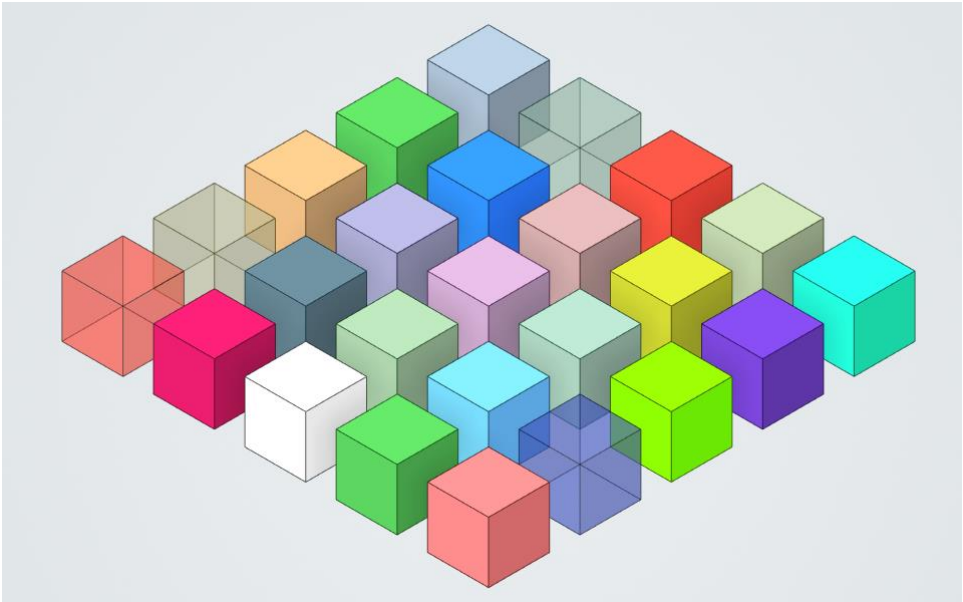


Figure 15: Body Colors Assigned by the User

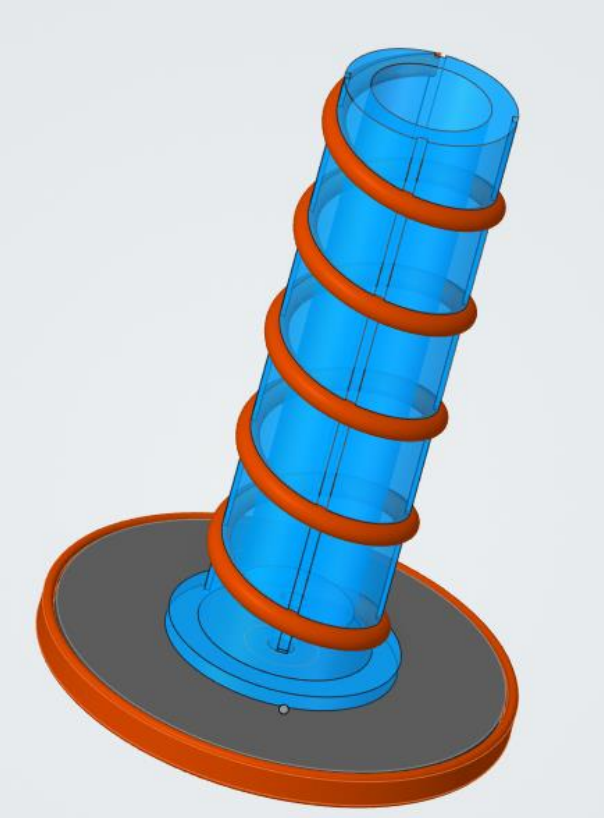


Figure 16: Body Colors Based on Assigned Materials

Lighting and Gradients

The shading of 3D icons and other 3D elements in the UI (including a 3D model, if relevant) are dictated by an imaginary light source placed at the top left. This position is based on the historical convention followed in Western art and studies in human perception.

The highlights are cast on the top and left-side features of the icons and shadows are cast on the bottom and right-side features of the icons. It is imperative to be consistent with the shading of 3D elements; otherwise, the user will be confused by the inconsistent visual perception of the scene.

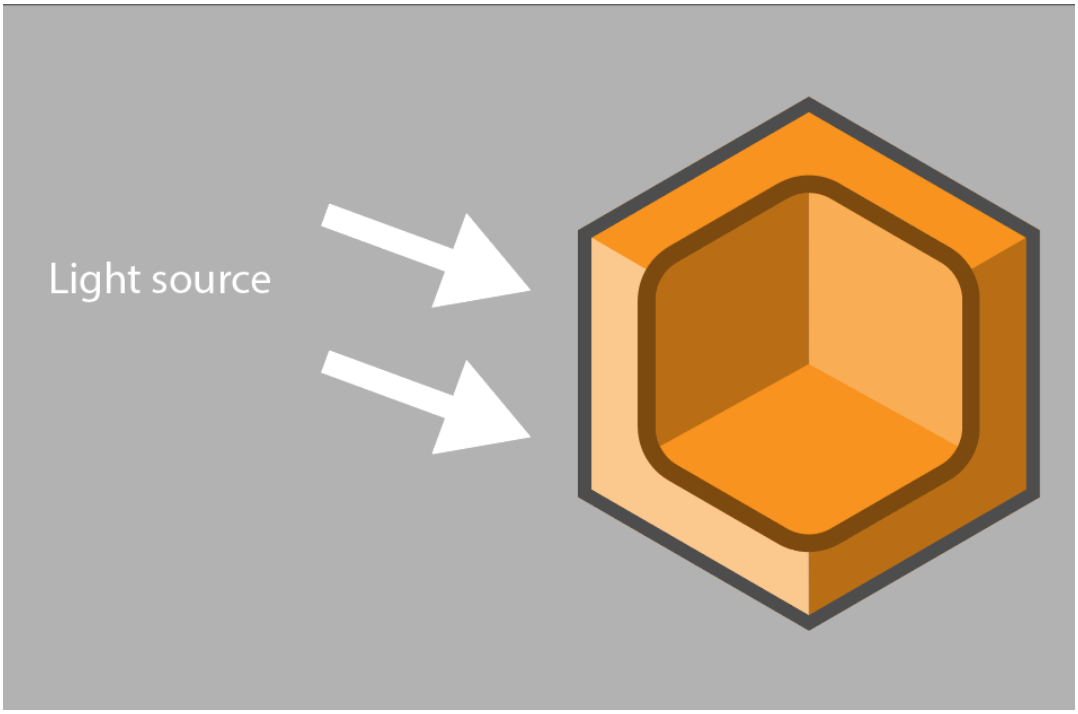


Figure 17: Effect of Light Source on Highlights and Shadows

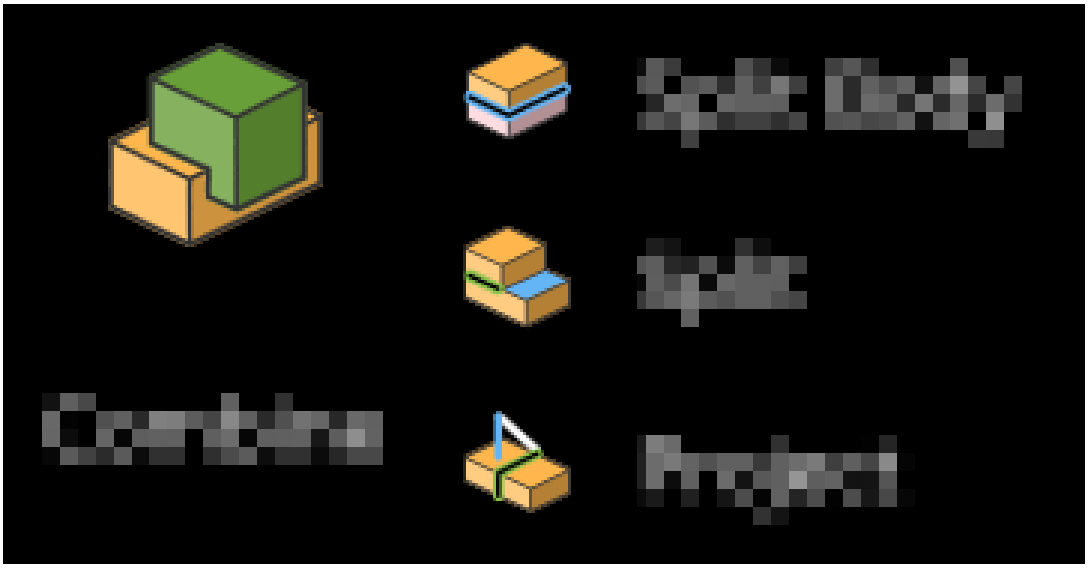


Figure 18: Consistent Highlight and Shadows Across a Group of Icons

In a 3D model-based application, slight gradient background coloring should be used for the scene in order to provide the necessary depth for an immersive experience. In other types of applications, a flat background color that contrasts well with the foreground UI elements should be used. (See Color Palettes and Their Usage for gradient and flat background colors.)

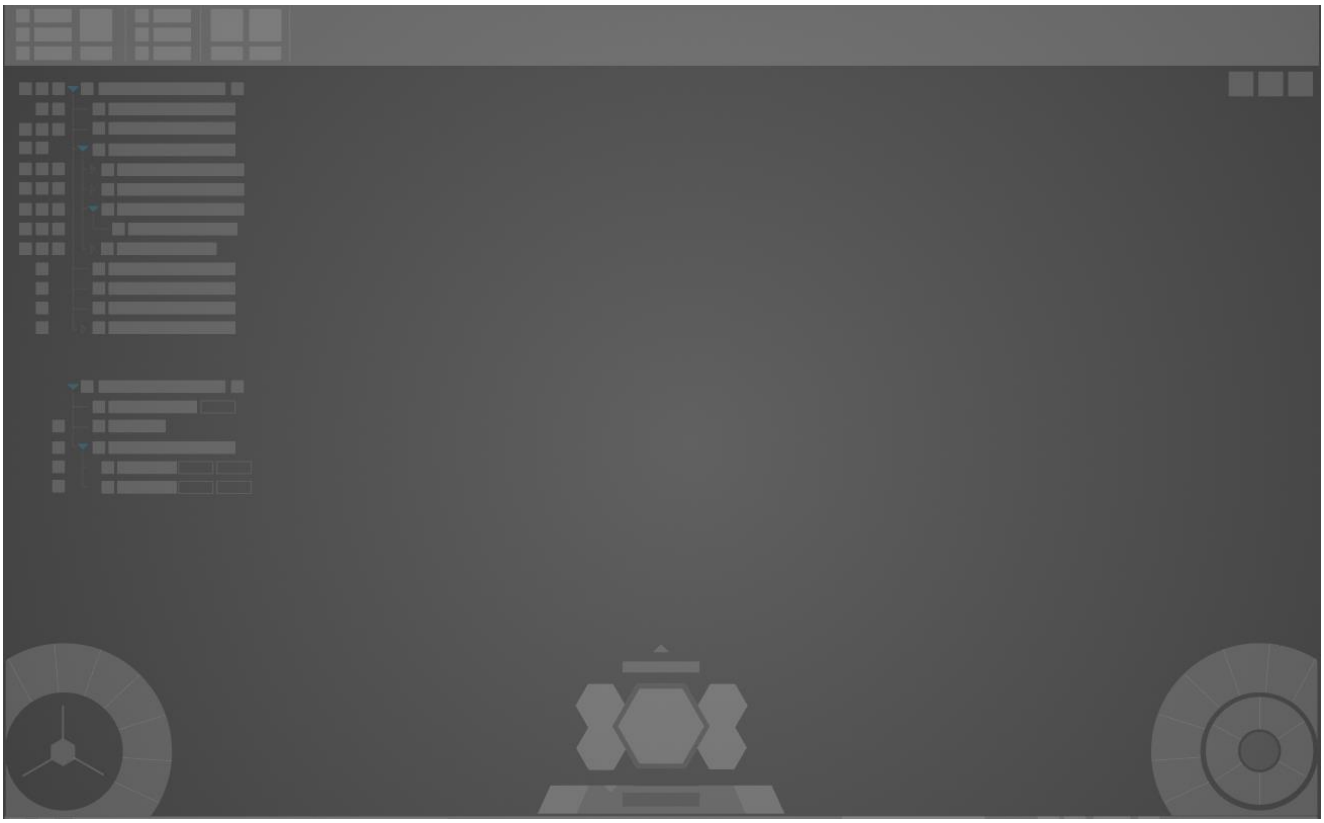


Figure 19: Gradient Background for 3D Model-Based Application

Themes

Theme options allow the user to choose the application-wide coloring (dark or light) best suited to their tastes or needs. Since this change affects the entire application, the choice of default theme should be part of the initial preference setup (user onboarding) on first usage of the product. If a product does not offer a user onboarding process, then the dark theme should be used by default. The theme selector also is typically found with other user preferences in the software's settings options. See Color Palettes and Their Usage for details about the dark and light theme palettes.

Dark Theme

In the dark theme, interface controls are subdued and elegant, allowing the content in the main viewing area, such as results of a 3D simulation, to stand out boldly as the application's centerpiece. This option is also easier on the eyes when viewing an application for a long period of time. The dark theme, however, does not lend itself to screen captures for use in printed documents.

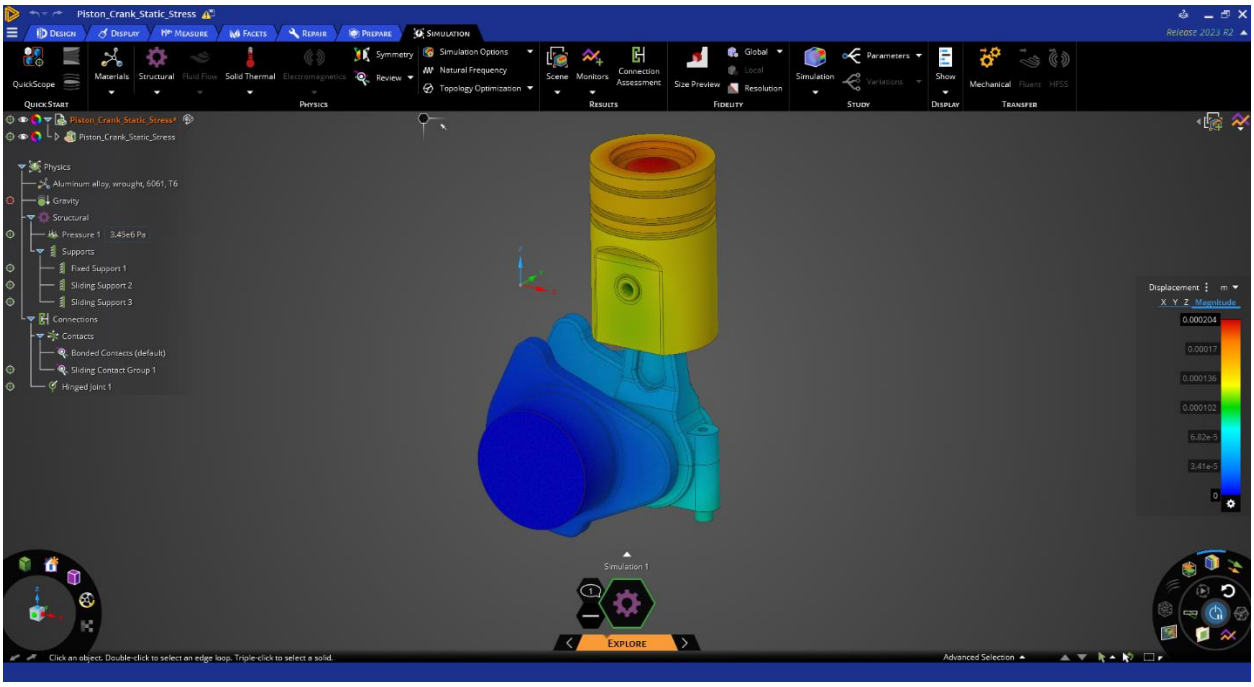


Figure 20: UI in the Dark Theme

Light Theme

Some users may prefer a brighter alternative. The light theme evokes the legacy interfaces of Ansys' products, offering a sense of familiarity to our users who are used to those applications and their coloration. It is also suitable for screen captures intended for use in printed documents. The light theme, however, can be harder on the eyes when viewing an application for a long period of time.

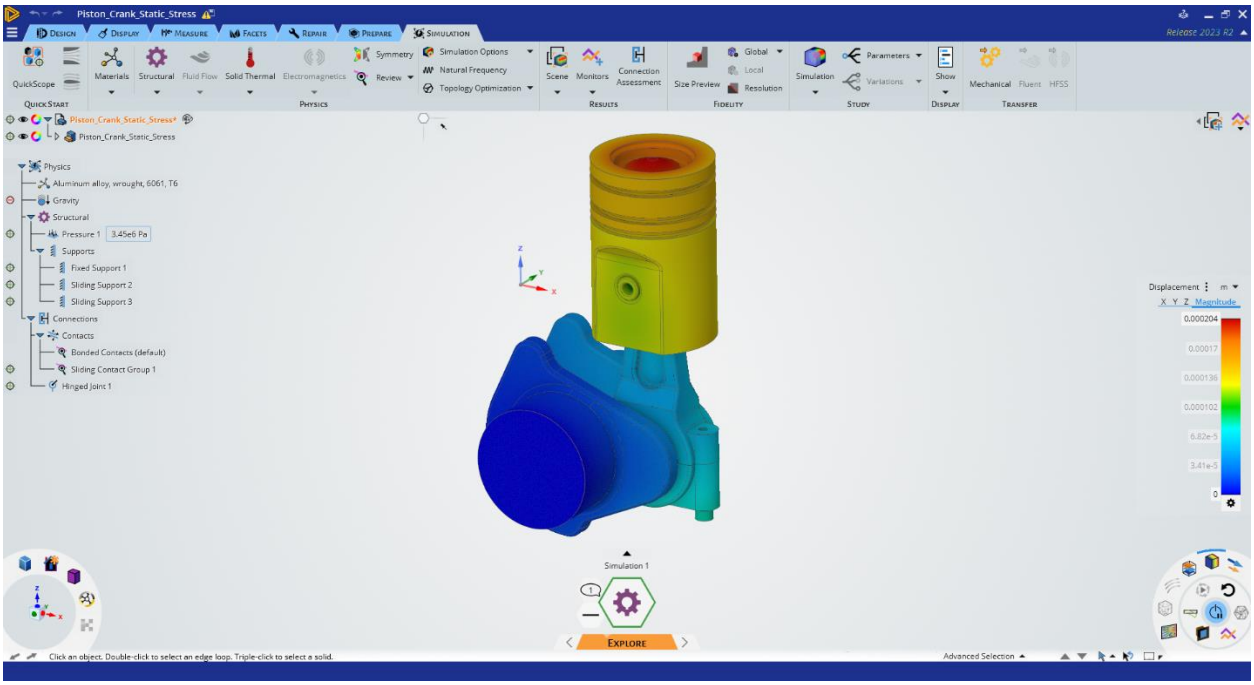


Figure 21: UI in the Light Theme

Typography

Typefaces and Fonts

Typefaces

Discovery uses four typefaces:

- **Alegreya Sans SC** (small caps) for titles and headings

  Alegreya Sans SC

  THE QUICK BROWN FOX

  JUMPS OVER THE LAZY DOG

- **Open Sans** for body text

  Open Sans

  The quick brown fox jumps over the lazy dog

- **Consolas** for Code console text

  Consolas

  The quick brown fox jumps over the lazy dog

- **Comic Neue** for help text

  Comic Neue

  The quick brown fox jumps over the lazy dog

Main UI Fonts

	Font	Size	Color	Examples
Body/Default	Open Sans Regular	12 px (1 rem)	Theme-dependent	Body text
Title	Alegreya Sans SC Medium	14 px (1.16 rem)	Theme-dependent	Ribbon heading, Panel title
Heading	Alegreya Sans SC Regular	12 px (1 rem)	Theme-dependent	Expander heading

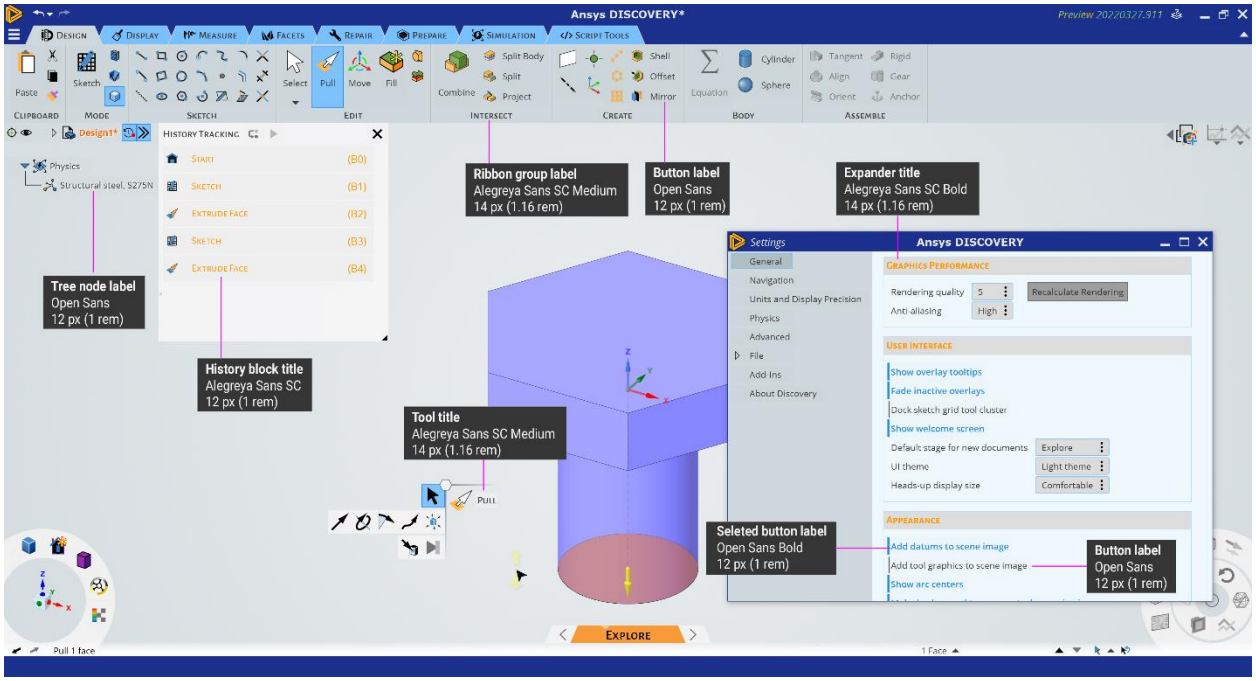


Figure 22: Font Specifications in the Main UI

Help Overlay Fonts

	Font	Size	Color	Examples
Body	Comic Neue Regular	12 px (1 rem)	Grey 1 #F5F5F5	Body text
Title	Comic Neue Bold	24 px (2 rem)	Orange 1 #FFA500	Main heading
Subtitle	Comic Neue Bold	14 px (1.16 rem)	Orange 1 #FFA500	Sub-heading

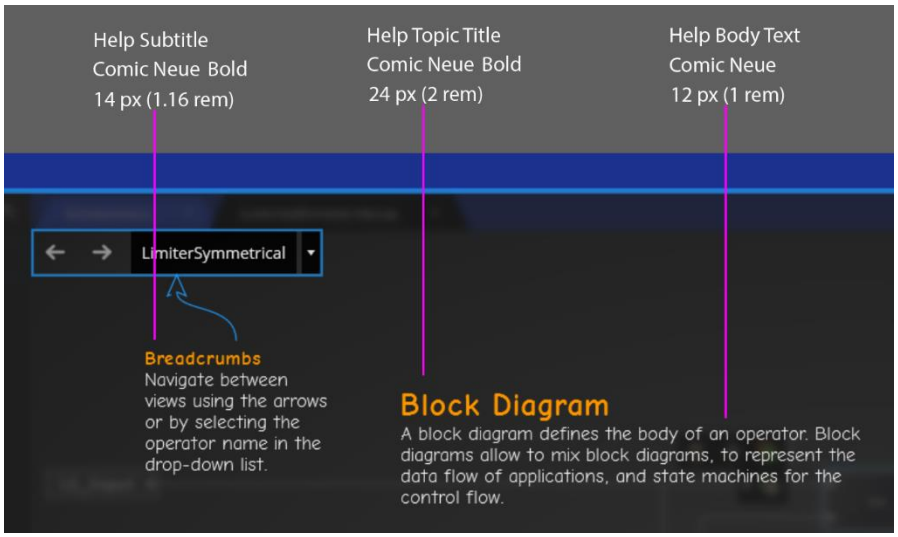


Figure 23: Font Specifications in the Overlay Help

Capitalization

Definitions

Capitalization styles can be used to aid in emphasis and clarity, but they can also slow comprehension if misapplied. There are three systems of capitalization commonly used for interface design:

- **Title-Style Capitalization**
  When using title-style capitalization, the first letter of each word is capitalized. Unless they begin the text, articles (a, an, the) and small coordinating words (and, or, of) are left lower-case.

- **Sentence-Style Capitalization**
  As the name implies, sentence-style capitalization follows the typical rules of a sentence. The first letter is capitalized, along with proper nouns or acronyms.

- **All Caps**
  Presenting text in all capital letters draws the eye and slows the reader.

Small caps is different from all caps because it is a typeface designed with the form of capital letters, but the height is taller for letters that are capitalized and shorter for those that are not. The capitalization styles above apply equally to small caps typefaces. See Typefaces for an example.

Title Style: When to Use

Title-style capitalization should be used for text that

- Represents a name, title, or group heading (e.g., page titles, tabs, icon labels, form labels)
- Labels a control that performs an action (e.g., context menus, drop-down menus, buttons)

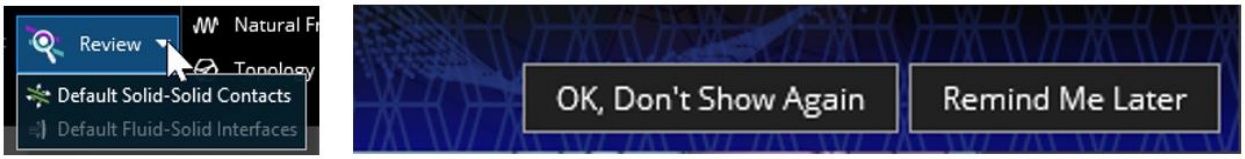


Figure 24: Examples of Title-Style Capitalization

Sentence Style: When to Use

Sentence-style capitalization is used for

- Anything that is meant to be read as a sentence.
  - Examples: body copy, tooltips, interface messages/notifications
- Labels of settings/options/inputs that do not perform actions.
  - Since there are often many of these shown together, and some are meant to be read as a sentence, using sentence-style capitalization ensures consistency within the group.
  - Examples: drop-down list labels, options in Settings panel, inputs in HUD or HUD options panel, options in panels that open from the ribbon.

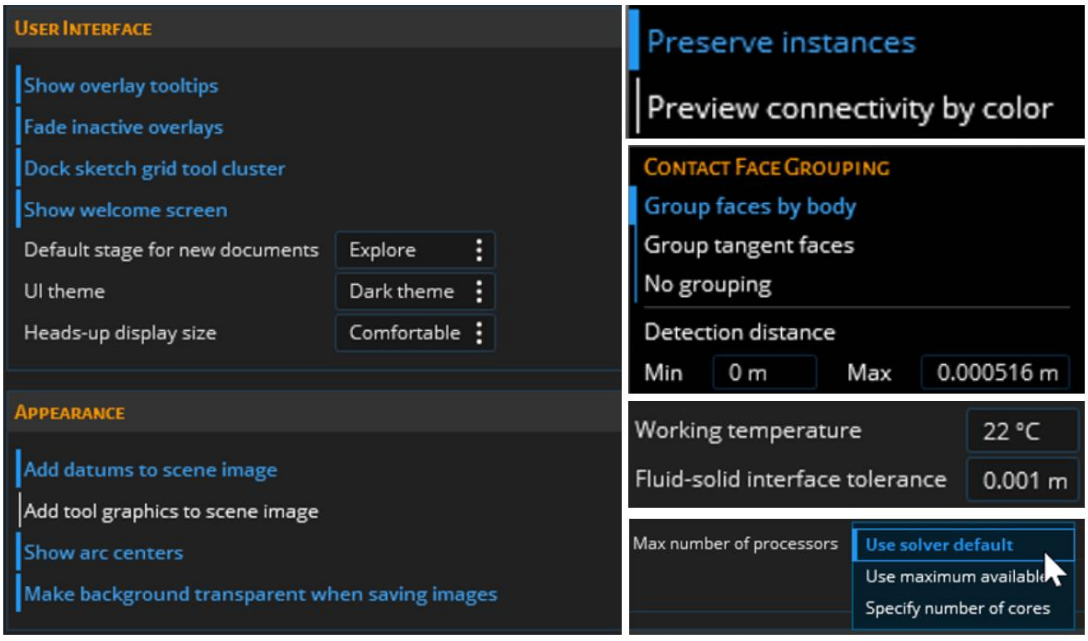


Figure 25: Examples of Sentence-Style Capitalization

Dealing with Conflicting Guidelines

If one of the items in a menu does not strictly perform an action, the title-style guideline will take precedence to ensure consistency within the menu. An example is Select Components in the menu below:

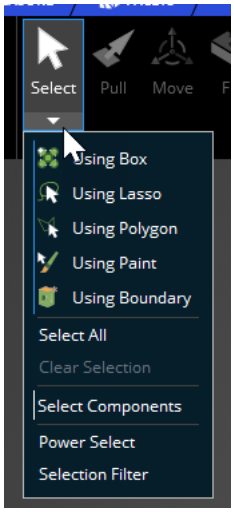


Figure 26: Ensuring Consistency in a Menu when Guidelines Conflict

All Caps: When to Use

All caps should be used for acronyms and for product names in the title bar (where appropriate based on branding). For other situations where all caps might be considered, use them sparingly and not for labels of more than a few words.

Small Caps: When to Use

As described in Definitions, small caps is a typeface rather than a style of capitalization. It should be used only for titles and headings. Examples include the choices on the left side of the HUD (since they are effectively sub-titles for the tool), header labels on expanders, and headings for groups of settings.

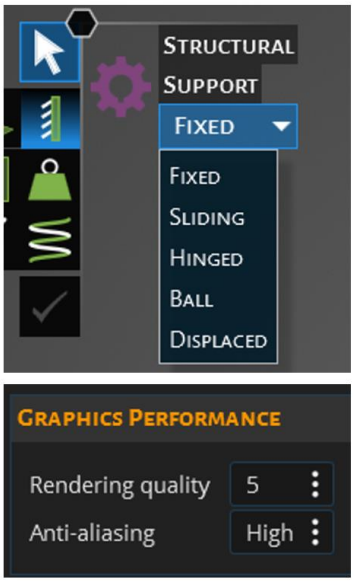


Figure 27: Examples of Small Caps Usage

User Entries

When a user enters text, do not apply capitalization transformations. Honor the formatting that the user intended.

Capitalization Based on Object Name

By default, object names are title case (e.g., Fixed Support 1, Max Displacement 3), but if the user changes the name, the capitalization for it will be whatever the user specifies (e.g., My distributed Force), regardless of where the name appears.

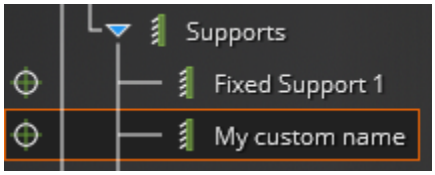


Figure 28: Capitalization of Tree Object with User-Edited Name

Capitalization Based on Outside Sources

Material assignments, for example, pick up capitalization from the material name, so capitalization will be dependent on the source of the material.

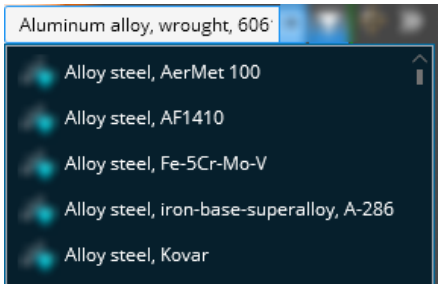


Figure 29: Capitalization of Material Names is Based on External Source

Punctuation

Labels should not end with a colon (:). Omitting the colon helps to reduce clutter on the screen.

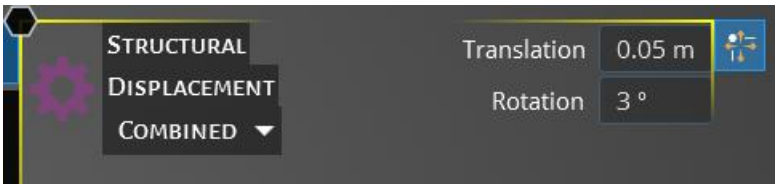
Font Treatments

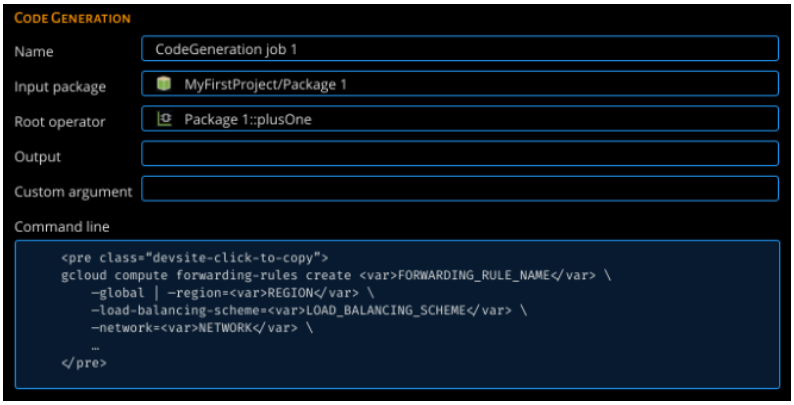
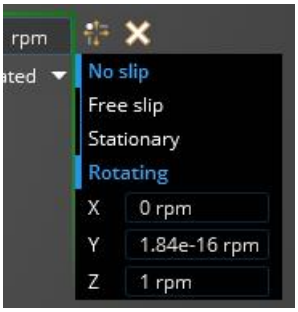
Font treatments such as italic and underline should be used sparingly and intentionally. Typically, underline is used to denote hyperlinks, and italic is used to denote emphasis or an out-of-date or in-progress state of displayed information. The meaning assigned to these special treatments should be clearly communicated to the user and consistently followed across the application.

Avoid using strikethrough treatment because it reduces the readability, does not look professional, and introduces clutter.

Field Label Placement

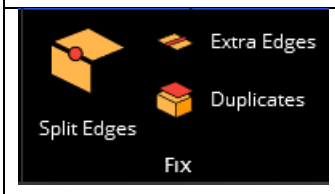
The placement of labels for input fields depends on the context. The overall goal is to balance the competing goals of easy scanning, effective space usage, and internationalization.

	For small input fields, such as in the HUD, labels are placed in front of the fields to make it easier to scan and identify the field of interest. On the left side of the HUD bracket, the labels are left-justified, while on the right side they are right-justified. This keeps the HUD more compact, and less likely to get in the way of what the user is viewing.

 	For multi-line input fields, labels should be placed above the fields for better sizing and layout management of the form or panel. In a form or panel, such as the HUD options, the labels and the corresponding fields should both be left-justified. Exceptions may be made if the labels in a group are of very different lengths, to avoid having any label in the group too far from its field. In forms where the user must fill in all fields, the labels are placed above the fields because there is no need to scan them. In this case, the internationalization guideline of placing labels above the fields takes priority.

Wrapping for Long Labels

For groups of buttons (e.g., in a ribbon), wrap long labels to optimize space usage for the set of buttons.

	The labels of the first two buttons here wrap to two lines because a single line would lead to extra blank horizontal space (if the other buttons shifted to the right) or cause the labels to touch other content (if the other buttons did not shift).
	Split Edges does not wrap because that would use too much vertical space, given that there are only two buttons stacked to the right of it, and there is adequate horizontal space between Split Edges and the adjacent buttons.

Icons

Icons are a powerful visual communication medium in the UI: using graphical elements, they convey the meaning of a button and its function; and the active and changed status of the application. Icons also contribute to conservation of screen real estate, making sure that every pixel counts. They also help users visually process concepts faster by making them recognize them, rather than recall.

Icon Guideline Cheat Sheet

For Designers: Creating Icons

We approach icon design by understanding why the user will look for an icon in the application, what it should represent, where it will display in the interface, and how it will render in different themes. Through our icons, we aim for consistency, legibility, and clarity.

Inspiration for icons may come from existing symbols and standards in simulation and physics software, mobile applications, games, and visual communication in the real world.

The process of icon design involves stakeholders at every stage:

1. Brainstorm ideas and create drafts as physical and/or digital sketches.
2. Select the best candidates and create mockups in different sizes and locations within the user interface. This helps us verify and validate our designs, in-context.
3. Use the in-context mockup to obtain feedback and iterate on the design until it satisfies stakeholder requirements.
4. Translate the final design to a vector-code format or image resource to be consumed by the product.

While the following rules and guidelines cater to most icons, thoughtful exceptions are allowed if they solve unique problems in visual communication.

To keep icons consistent, clear, flexible, and easily scalable, follow these rules, guidelines, and tips.

Rules, Guidelines, and Tips

Rules

- Icon content preparation
    - All content must be within a square canvas of side 500px.
    - The outer stroke must be 12px wide and the internal stroke 10px wide.
    - Contents should touch at least two opposite edges of the bounding square canvas.
        - Edge to edge allows the icons to be on the same scale and reduce variability in terms of different padding
- Shape and color
    - The final icon must consist of only closed vector paths and shapes.
        - A closed path is continuous and has no beginning or end; for example, a circle is a closed path. In contrast, an open path has a beginning point that is not the same as the endpoint; for example, a straight line is an open path.

        - It is always advisable to close your paths when creating shapes so there will be no problems with the fill being misinterpreted as you export the icons to other applications.
        - Learn more about this topic.
    - Colors used must be from the appropriate palette.
    - Icons must have a semi-flat (2.5D) look; use gradients only as appropriate (for example, to show result contours or curved surfaces).
    - Symbols must be used consistently across icons.

Guidelines

- Always design icons by keeping in mind the context of use, including whether it is important for an icon's colors to be the same across themes (see Themes and Icon Colors).
- Use the minimum number of colors needed to get the meaning across.
- Constrain yourself to the smallest size the icon will show up in the UI. It is always easy to scale up rather than down.
- Be mindful of negative space and overlap.
- Use a design grid for consistent sizing.
- Use glow to enhance visibility against dark backgrounds.

Tips

- Try out the icons with exact/intended colors over samples of UI at the original button size using pixel previewing in your design tool (e.g., the Pixel Preview feature in Adobe Illustrator).
- Save different versions and iterations of the icon as changes occur.

Additional Information

Semi-Flat Styling

Icons have a semi-flat look, which means that while lighting and shadows make them appear somewhat 3D, they do not follow the rules of rendering like a real physical object. So, multiple elements in these icons do not cast shadows upon each other. The styling of these icons is inspired from marker rendering and painting techniques.

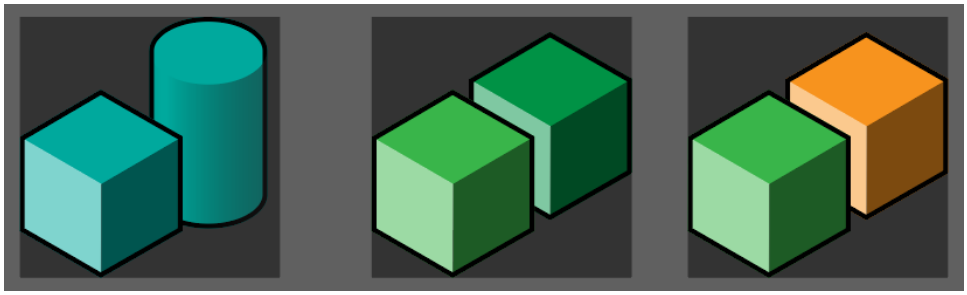


Figure 30: Examples Showing Effect of Semi-Flat Styling on Multiple Elements

Sizes

These are the sizes to use for icons in a product UI:

General sizes: **8px, 16px, 24px, 28px, 32px, 48px, 64px, 96px**

Cursor sizes: **12px, 24px, 32px**

Using a Grid

We use a grid (provided in the separate file named Grid for icon design.svg) for balance and visibility. But this is only to be used as a starting reference, and not as a strict outline for placing shapes within the square artboard.

Deviating from the grid may be necessary for clarity and context. Sometimes the aesthetics override the structure of the grid: the contents of the icon itself might not fit properly with the grid. The interplay of different elements in the icon is also important, as in the example below where the change in thickness of an element "breaks" the grid.

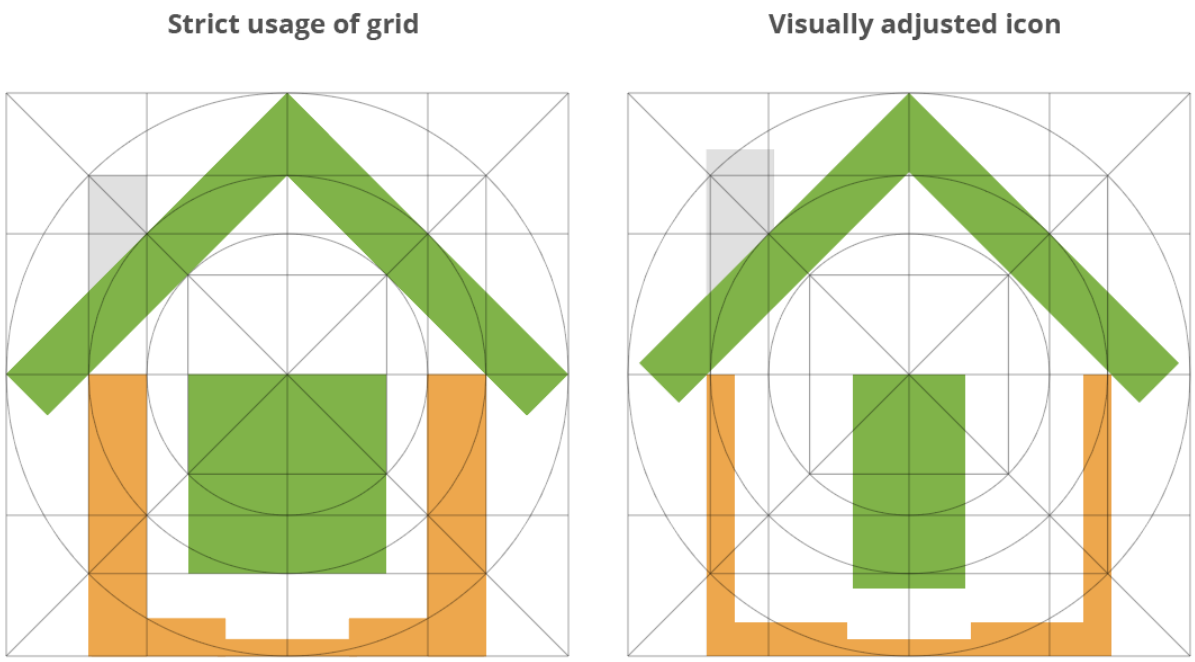


Figure 31: Example of Deviating from the Grid to Improve an Icon

Themes and Icon Colors

Theme-specific icons have some, or all, shapes that use different colors based on theme. (See Theme-Dependent Coloring for details.) There are six theme-dependent colors (refer to the cheat sheet for more information) to be used.

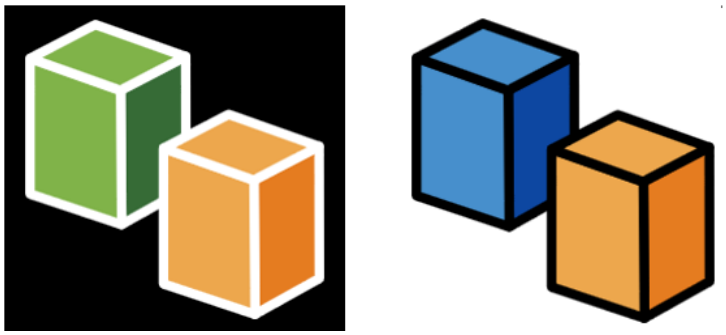


Figure 32: Theme-Specific Versions of an Icon Shown in Dark Theme (Left) and Light Theme (Right)

Static icons have shapes that must not change color based on theme. These icons must look consistent across the themes, just like branding icons. Static coloring is appropriate when the colors have some special significance in the product. In the example below, the magenta and orange colors are associated with different simulation modes.

Figure 33: Same Static Icon Shown in Dark Theme (Left) and Light Theme (Right)

Adding Glow

Icons, particularly of small size, might have outlines with decreased visibility against dark backgrounds.

To make the icon outline legible against the dark background, add an additional offset outline of white (#FFFFFF) set at 75% opacity. This renders a small “glow” around the icon when placed on a dark background, creating a visual boundary between its outline and the similar-colored background. The glow is not noticeable in the light background.

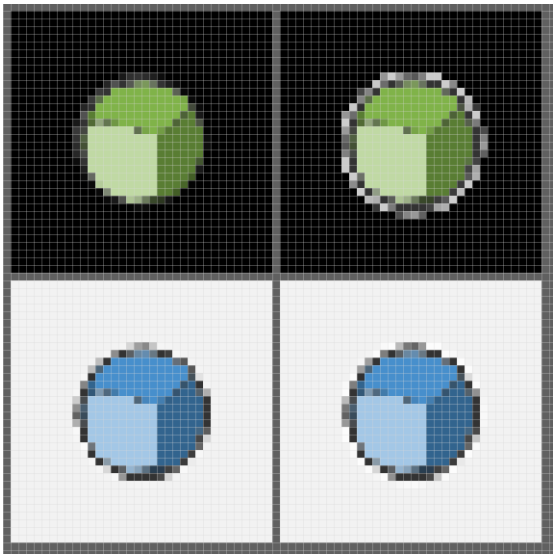


Figure 34: Pixel-Level View of Adding Glow on Dark and Light Backgrounds

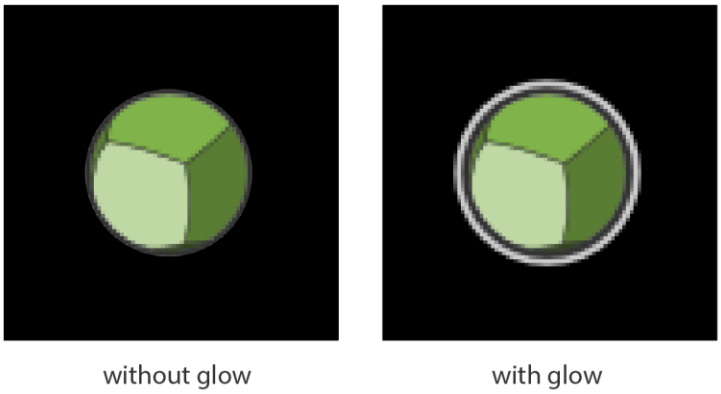


Figure 35: Dark Background: Icon with Glow Has a Visible Outline

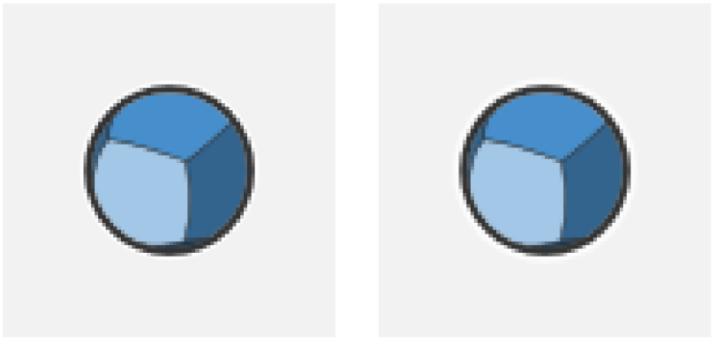


Figure 36: Light Background: Glow is Not Noticeable

Pixel Pushing

Contents of some icons might not render correctly at a small size. This is usually caused by the individual pixels of the icon shapes not aligning to the display's pixel grid.

To address this problem, tweak the icon's vector shapes while also checking how they render as pixels within a small size canvas (usually 16px).

If the icon will be used in the UI only in one specific size, the design should primarily address that size, but it is good practice to design for all the possible sizes. The pixel pushing is most useful for small size icons; for larger size icons, this step is recommended but not required.

Taking the example of a 12px icon, we can see that the one on the right has sharper details.

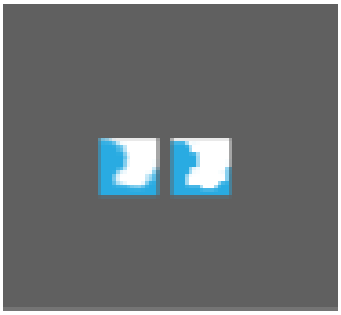


Figure 37: Icons without Pixel Pushing (Left) and with Pixel Pushing (Right)

Looking at these icons closely, notice that the elements are different. The icon on the right renders more crisply in smaller sizes because the colors are not partially allotted to a pixel (which leads to blur and fuzziness).

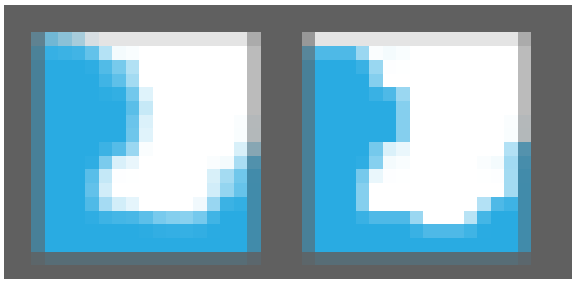


Figure 38: Zoomed-In View of Icons Shown Above

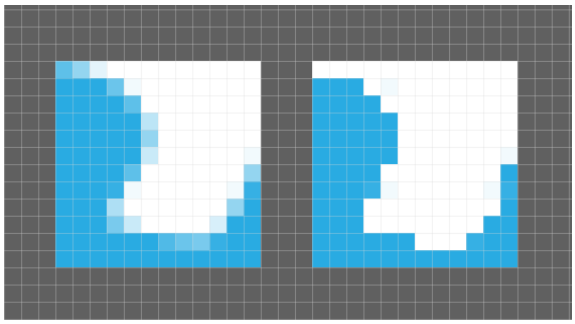


Figure 39: Pixel Grid Overlaid on Icons

Using Symbology

To make specific actions intuitive, use a standard set of symbols and graphic elements across icons. These symbols should convey functional or domain-specific engineering simulation concepts, and they are typically held constant within and across icon category groupings.

Examples:

- Plus (+) symbols indicate a positive change in the application, like creating or adding something to a model or a physics condition.
- The "X" symbol is often colored red to indicate negative changes such as canceling a process or removing a component.
- Asterisks (*) are often colored orange to indicate a reference or establish a relationship between processes or components in the application.

Symbols can be placed on a scale that ranges from completely abstract to the exact representation of the idea itself. Examples are shown below.

Figure 40: Custom Abstract Symbols

Figure 41: Universally Understood Abstract Symbols

Figure 42: Brand Logos

Figure 43: Representational Symbols of Objects

Figure 44: Combining Symbols to Generate Meaning

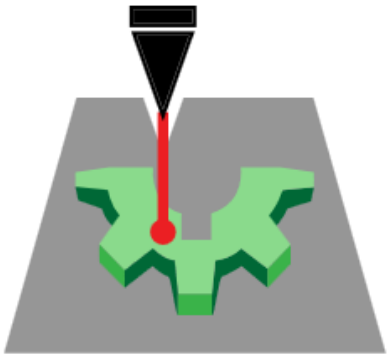
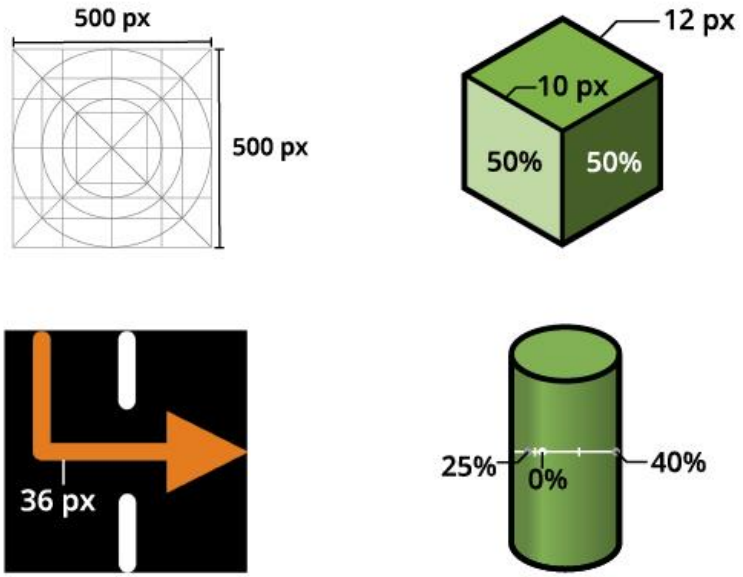


Figure 45: Representations of Objects

Cheat Sheet



Usage	Dark Theme	Light Theme
Primary Most contrast with the background	#FFFFFF	#000000
Primary Shadow A shade of primary color that provides less contrast	#E0E0E0	#727272
Accent Most used icon color in the theme. Use this as the first option to add a color to your icon	#80B349	#478FCC
Accent Shadow A shade of accent color to be used when a darker tone is needed, usually to represent shadow cast surfaces	#366B36	#0D47A1
Second Accent Use this color when a secondary color is needed in the icon	#EDA74D	
Second Accent Shadow Use this color when a darker shade of the second accent color is needed	#E57C20	

Figure 46: Cheat Sheet for Icon Design