

# Package ‘ganttify’

October 29, 2025

**Type** Package

**Title** Create Interactive Gantt Charts with Work Breakdown Structure

**Version** 0.1.6

**Description** Create Primavera-style interactive Gantt charts with Work Breakdown Structure (WBS) hierarchy and activities. Features include color-coded WBS items, indented labels, scrollable views for large projects, dynamic date formatting, and the ability to dim past activities. Built on top of 'plotly' for interactive visualizations.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5.0)

**Imports** plotly (>= 4.9.0), dplyr (>= 1.0.0), htmlwidgets (>= 1.5.0),  
magrittr, rlang

**RoxygenNote** 7.3.3

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**URL** <https://github.com/AhmedAredah/Ganttify>

**BugReports** <https://github.com/AhmedAredah/Ganttify/issues>

**NeedsCompilation** no

**Author** Ahmed Aredah [aut, cre]

**Maintainer** Ahmed Aredah <Ahmed.Aredah@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-10-28 23:10:02 UTC

## Contents

Ganttify . . . . .	2
test_project . . . . .	6
<b>Index</b>	<b>9</b>

## Description

Creates a Primavera-style interactive Gantt chart with Work Breakdown Structure (WBS) hierarchy and activities. The chart features color-coded WBS items, indented labels, scrollable view for large projects, and dynamic date formatting.

## Usage

```
Ganttify(
  wbs_structure,
  activities,
  chart_title = "Project Gantt Chart with WBS",
  x_range = NULL,
  milestone_lines = NULL,
  color_config = NULL,
  display_config = NULL,
  label_config = NULL,
  bar_config = NULL,
  layout_config = NULL
)
```

## Arguments

- |               |  |
|---------------|--|
| wbs_structure | A data frame with 3 columns: ID (character), Name (character), and Parent (character). Parent should be "None" or "" for root level items.   |
| activities    | <p>A data frame with 5 required columns, 2 optional columns, and any number of additional columns:</p> <ul style="list-style-type: none"> <li>• WBS_ID (character): Associated WBS item identifier</li> <li>• Activity_ID (character): Unique activity identifier</li> <li>• Activity_Name (character): Activity name</li> <li>• Start_Date (character): Planned start date in MM/DD/YYYY format</li> <li>• End_Date (character): Planned end date in MM/DD/YYYY format</li> <li>• Start_Date_Actual (character, optional): Actual start date in MM/DD/YYYY format</li> <li>• End_Date_Actual (character, optional): Actual end date in MM/DD/YYYY format. If Start_Date_Actual is provided but End_Date_Actual is missing, the actual bar will show from Start_Date_Actual to today (if today &gt; Start_Date_Actual).</li> <li>• Additional columns (optional): Any extra columns (e.g., Status, Agency, Priority) are preserved and can be used for attribute-based coloring via color_config with mode="attribute".</li> </ul> |

	When actual dates are provided, activities display as stacked bars: planned on top (solid color) and actual on bottom (diagonal stripe pattern).
chart_title	Character. Title displayed at the top of the chart. Default "Project Gantt Chart with WBS".
x_range	Character vector. Date range for x-axis zoom (e.g., c("2024-01-01", "2024-12-31")). If NULL, shows full project range.
milestone_lines	<p>Data frame or NULL. Optional milestone lines to display on the chart. If provided, must be a data frame with the following columns:</p> <ul style="list-style-type: none"> <li>• date (required): Date for the vertical line (character in MM/DD/YYYY format or Date object)</li> <li>• label (required): Text label to display on the line</li> <li>• color (optional): Line color (e.g., "red", "#FF0000"). Defaults to color palette if not provided.</li> <li>• dash (optional): Line style - "solid", "dash", "dot", or "dashdot". Default "dash".</li> <li>• width (optional): Line width in pixels. Default 2.</li> <li>• label_position (optional): Label position - "top", "middle", or "bottom". Default "top".</li> </ul> <p>Default NULL (no milestone lines).</p>
color_config	<p>List or NULL. Configuration for chart colors. Structure depends on mode:</p> <ul style="list-style-type: none"> <li>• mode="wbs" (default if NULL): Activities inherit colors from parent WBS  <code>list(mode = "wbs", wbs = list("W1" = "#FF6B6B", "W2" = "#4ECDC4"))</code></li> <li>• mode="uniform": All activities same color, WBS same color  <code>list(mode = "uniform", uniform = list(wbs = "#34495E", activity = "#2ECC71"))</code></li> <li>• mode="attribute": Color activities by attribute column (e.g., Status)  <code>list(mode = "attribute",  attribute = list(column = "Status",  mapping = list("completed" = "green", "in-progress" = "orange"),  wbs = "#34495E"))</code></li> </ul> <p>If NULL, uses mode="wbs" with default color palette. Default NULL.</p>
display_config	<p>List or NULL. Controls visibility of chart elements. Structure:</p> <ul style="list-style-type: none"> <li>• wbs: List with show (logical), show_labels (logical), show_names_onBars (logical)</li> <li>• activity: List with show (logical), show_names_onBars (logical)</li> </ul> <p>Example:</p> <pre>list(   wbs = list(show = TRUE, show_labels = TRUE, show_names_onBars = TRUE),   activity = list(show = TRUE, show_names_onBars = FALSE) )</pre> <p>If NULL, uses defaults shown above. Default NULL.</p>
label_config	List or NULL. Template strings for labels. Structure:

- activity: List with yaxis (template for y-axis labels) and bar (template for bar labels)
- wbs: List with yaxis and bar templates

Available placeholders for activity: name, id, start, end, start\_actual, end\_actual, duration, wbs\_id (use with curly braces) Available placeholders for wbs: name, id, start, end, duration (use with curly braces) Example:

```
list(
  activity = list(yaxis = "{name} ({start} - {end})", bar = "{name}"),
  wbs = list(yaxis = "{name}", bar = "{name}")
)
```

If NULL, uses default template for all labels. Default NULL.

bar\_config

List or NULL. Styling configuration for bars. Structure:

- wbs: List with opacity (0-1) and height (numeric)
- activity: List with opacity (0-1), height (numeric), dim\_opacity (0-1), and dim\_past\_activities (logical)

The dim\_past\_activities field controls whether activities that end before today are dimmed. When TRUE, completed activities use the dim\_opacity value instead of the regular opacity. Example:

```
list(
  wbs = list(opacity = 0.3, height = 0.3),
  activity = list(opacity = 1.0, height = 0.8, dim_opacity = 0.3, dim_past_activities = F)
)
```

If NULL, uses defaults shown above. Default NULL.

layout\_config

List or NULL. Chart layout settings. Structure:

- buffer\_days: Numeric, days to add before/after timeline for margin
- indent\_size: Numeric, spaces per indentation level
- max\_visible\_rows: Numeric, maximum visible rows (enables scrolling)
- y\_scroll\_position: Numeric or NULL, initial scroll position
- yaxis\_label\_width: Numeric, width of y-axis label area in pixels (default 300)
- yaxis\_label\_max\_chars: Numeric or NULL, maximum characters for labels before truncating with "..." (NULL = no truncation)
- hover\_popup\_max\_chars: Numeric, maximum characters per line in hover popups before wrapping to next line (default 50)

Example:

```
list(
  buffer_days = 30,
  indent_size = 2,
  max_visible_rows = 20,
  y_scroll_position = NULL,
  yaxis_label_width = 300,
  yaxis_label_max_chars = NULL,
  hover_popup_max_chars = 50
)
```

If NULL, uses defaults shown above. Default NULL.

**Value**

A plotly object containing the interactive Gantt chart. Can be displayed directly or saved using `htmlwidgets::saveWidget()`.

**Examples**

```
# Load test data
data(test_project)

# Basic Gantt chart with WBS colors
chart <- Ganttify(
  wbs_structure = test_project$wbs_structure,
  activities = test_project$activities,
  color_config = list(mode = "wbs", wbs = test_project$colors)
)
chart

# Uniform color mode
chart <- Ganttify(
  wbs_structure = test_project$wbs_structure,
  activities = test_project$activities,
  color_config = list(
    mode = "uniform",
    uniform = list(wbs = "#34495E", activity = "#2ECC71")
  )
)
chart

# Attribute-based coloring (requires extra column in activities)
# Add a Status column to activities dataframe
activities_with_status <- test_project$activities
activities_with_status$Status <- sample(c("completed", "in-progress", "pending"),
                                       nrow(activities_with_status), replace = TRUE)

chart <- Ganttify(
  wbs_structure = test_project$wbs_structure,
  activities = activities_with_status,
  color_config = list(
    mode = "attribute",
    attribute = list(
      column = "Status",
      mapping = list("completed" = "green", "in-progress" = "orange", "pending" = "gray"),
      wbs = "#34495E"
    )
  )
)
chart

# WBS-only view using display_config
chart <- Ganttify(
  wbs_structure = test_project$wbs_structure,
  activities = test_project$activities,
  display_config = list(activity = list(show = FALSE))
)
```

```

)
chart

# Custom labels showing date ranges
chart <- Ganttify(
  wbs_structure = test_project$wbs_structure,
  activities = test_project$activities,
  label_config = list(
    activity = list(yaxis = "{name} ({start} - {end})")
  )
)
chart

# Customize bar heights and enable dimming for past activities
chart <- Ganttify(
  wbs_structure = test_project$wbs_structure,
  activities = test_project$activities,
  bar_config = list(
    wbs = list(opacity = 0.5, height = 0.4),
    activity = list(height = 0.9, dim_past_activities = TRUE, dim_opacity = 0.4)
  )
)
chart

# Add "today" line as a milestone
chart <- Ganttify(
  wbs_structure = test_project$wbs_structure,
  activities = test_project$activities,
  milestone_lines = data.frame(
    date = Sys.Date(),
    label = "Today",
    color = "red"
  )
)
chart

# Narrow label area with truncation
chart <- Ganttify(
  wbs_structure = test_project$wbs_structure,
  activities = test_project$activities,
  layout_config = list(
    yaxis_label_width = 200,
    yaxis_label_max_chars = 25
  )
)
chart

```

## Description

A sample project dataset for testing and demonstrating the Ganttify package. Contains a complete Work Breakdown Structure (WBS), activities, and custom colors for an example software development project.

## Usage

```
test_project
```

## Format

A list with 3 components:

**wbs\_structure** A data frame with 8 rows and 3 columns:

- ID: WBS item identifier (W1-W8)
- Name: WBS item name (e.g., "Project Summary", "Design Phase")
- Parent: Parent WBS ID or "None" for root items

**activities** A data frame with 15 rows and 7 columns:

- WBS\_ID: Associated WBS item identifier
- Activity\_ID: Activity identifier (A1-A15)
- Activity\_Name: Activity name (e.g., "Design UI", "Code Frontend")
- Start\_Date: Planned start date in MM/DD/YYYY format
- End\_Date: Planned end date in MM/DD/YYYY format
- Start\_Date\_Actual: Actual start date in MM/DD/YYYY format (some NA for not started)
- End\_Date\_Actual: Actual end date in MM/DD/YYYY format (some NA for in-progress)

Includes examples of on-time, delayed, ahead-of-schedule, and in-progress activities.

**colors** A named list of 8 colors:

- Each WBS item (W1-W8) is assigned a custom hex color

## Source

Example software development project

## Examples

```
# Load the test data
data(test_project)

# View structure
str(test_project)

# Create a Gantt chart

chart <- Ganttify(
  wbs_structure = test_project$wbs_structure,
  activities = test_project$activities,
  color_config = list(mode = "wbs", wbs = test_project$colors)
```

)  
chart

# Index

\* **datasets**

test\_project, [6](#)

Ganttify, [2](#)

test\_project, [6](#)