
Building Custom Visuals for Pyramid Analytics with Claude

**A step-by-step guide for creating visuals without
writing code yourself**

Who this guide is for

This guide is for anyone who wants to build a Custom Visual in Pyramid Analytics and doesn't write JavaScript code. You describe what you want in plain language, Claude writes the visual script for you, and you paste the result into Pyramid. No programming knowledge required.

Along the way you'll pick up just enough of the basic structure to recognize a working result and describe changes with confidence — nothing more technical than that.

Before you start

You'll need:

- Access to Pyramid Analytics with permission to create Custom Visuals
- A clear idea of what you want the visual to look like and do (sketches, examples, or reference images all help)
- Access to Claude

Step 1. Set up the Custom Visual shell

Every Custom Visual in Pyramid starts from the same basic structure, sometimes called the “shell.” Think of it like a picture frame: it doesn't matter what picture goes inside, but the frame always has the same basic parts.

When you ask Claude to build a visual, it always sets up this same starting structure behind the scenes, so you don't have to think about it.

Start with guiding Claude on what you need and how to output the script file:

1. Tell Claude that you want to create a new visual and provide it with relevant references to Pyramid's Custom Visual APIs:

“I want to build a new Custom Visual for Pyramid Analytics. Here are some relevant links...”

For a list of links - refer to Appendix A at the end of the document
2. Output a **single complete JavaScript file** — no HTML wrapper, no `<script>` tags.
3. Start with a **header comment block** describing the visual, drop zones, and any configuration notes.
4. Follow with the **CONFIG block**, then `main()`, then `render()`, then helpers.
5. The CONFIG block should allow for modification of the visual by the user:
 - a. Font properties
 - b. Canvas colors
 - c. Elements size, colors, fonts etc.

NOTE: The Custom Visual is not controlled through the Formatting Panel of the Discovery, therefore, any formatting control required for your visual needs to be specified to Claude to be added to the Config section.

6. After the code, provide a **short usage guide** explaining:
 - a. What to put in each drop zone
 - b. Which CONFIG values to tweak for common customizations
 - c. Any known limitations or edge cases

Good to know: You never need to write or edit this structure yourself. Claude handles it automatically every time it builds a visual for you — this section is just so the result looks familiar when you see it.

Step 2. Describe the visual to Claude

The clearer your description, the closer Claude's first attempt will be to what you want. A strong description covers four things:

1. What the visual should look like
2. What data it needs to show
3. What should happen when someone clicks or interacts with it
4. Any special requirements (for example, support for right-to-left languages, or a specific number of levels in a hierarchy)

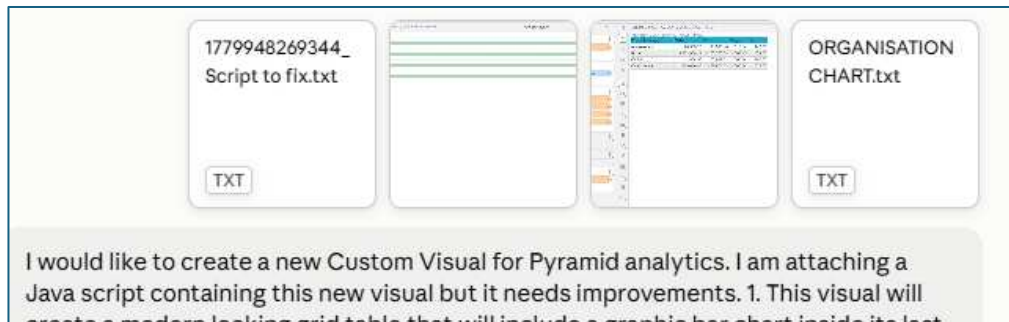
Example prompt

"Build a Pyramid Custom Visual: an expandable org chart. Each box should show a person's name and a small trend line. Clicking a box should highlight it and update the rest of the dashboard. Support up to four levels deep."

Help Claude do a better job

You don't have to describe everything in words. Claude can also work from things you show it:

- **Upload an image** — a screenshot, mockup, hand-drawn sketch, or an example from another tool. Claude can use it as a visual reference for layout, colors, or style.
- **Share an existing Custom Visual code** — if you already have a similar Custom Visual from a past project, share that file with Claude. It can reuse the parts that work and adjust the rest to match your new request.
- **Combine both** — for example, an image showing the look you want alongside an old visual's code for reference on behavior. The more context Claude has, the fewer rounds of back-and-forth you'll need.
- **Include a Discovery Grid image:** Build a grid with the actual data from your model and take a snapshot of it. This will show Claude where your actual data will reside and will help it better utilize the Rows, Columns, Values, Size, etc. drop zones.



Example of images and text file uploaded to Claude for creating a new visual

Step 3. Review what Claude gives you

Claude will reply with a complete file and a short usage note underneath explaining:

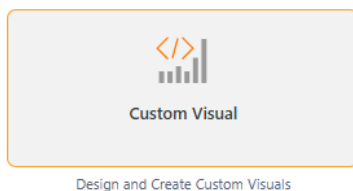
- What to place in each section of the data (for example, which field goes where)
- Which settings you can safely adjust yourself, like colors or sizes
- Anything the visual can't currently do

You don't need to read or understand the file itself — treat it like a finished document you're about to hand off. The usage note is the part worth reading.

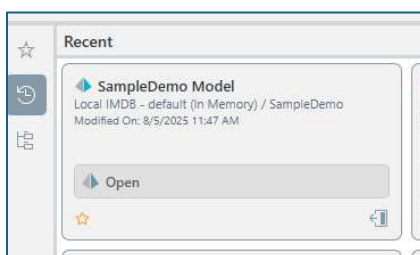
Step 4. Load it into Pyramid

Now bring the visual into Pyramid:

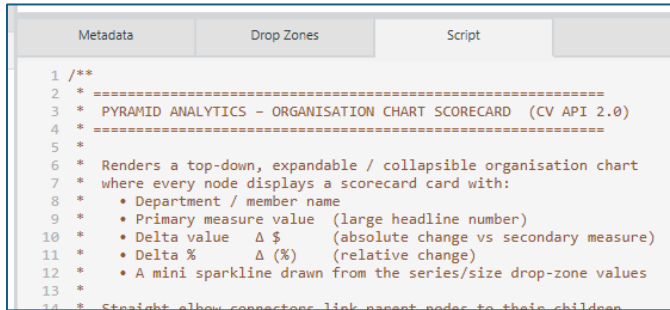
1. Open New Formulation > Custom Visual



2. Select the Model it works with



- Paste the entire script file Claude gave you into the Script tab

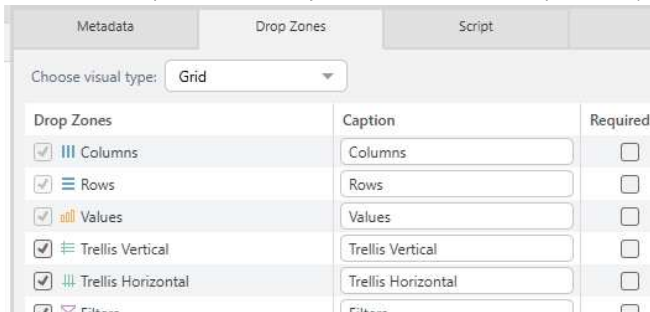


```

1 /**
2  *
3  * PYRAMID ANALYTICS - ORGANISATION CHART SCORECARD (CV API 2.0)
4  *
5  *
6  * Renders a top-down, expandable / collapsible organisation chart
7  * where every node displays a scorecard card with:
8  *   • Department / member name
9  *   • Primary measure value (large headline number)
10 *   • Delta value Δ $ (absolute change vs secondary measure)
11 *   • Delta % Δ (%) (relative change)
12 *   • A mini sparkline drawn from the series/size drop-zone values
13 *
14 * Straight elbow connectors link parent nodes to their children

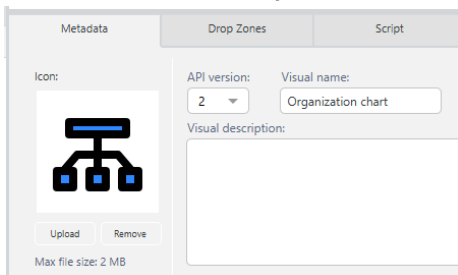
```

- Define which Drop Zone to show for this visual
You can keep them all or just leave those required by the visual's code.



Drop Zones	Caption	Required
☒ Columns	Columns	☐
☒ Rows	Rows	☐
☒ Values	Values	☐
☒ Trellis Vertical	Trellis Vertical	☐
☒ Trellis Horizontal	Trellis Horizontal	☐

- Fill the Metadata tab: Give it a name and upload SVG image
You can find SVG images online



Icon:  Upload Remove

API version: 2 Visual name: Organization chart

Visual description:

Max file size: 2 MB

- Click "Apply" to update the changes and then Save the visual into Pyramid.
- The visual will be available when opening a Discovery from the selected model under the Custom Visual menu:



Step 5. Fix problems by going back to Claude

If something doesn't look right, the fastest path is to go back to the same Claude conversation and describe exactly what you see. A specific description gets a specific fix much faster than a general one.

What to tell Claude

- What you expected to happen versus what actually happened.
- Whether it looks wrong immediately, or only after you click something
- A screenshot of the problem, if you can take one

Common issues and what they usually mean

What you see	Likely cause
The visual is blank	A required field hasn't been placed in the right section yet
Clicking does nothing	The interactive part of the visual needs a small adjustment — describe what you clicked and what you expected
The view resets or jumps after clicking	The visual isn't remembering your scroll position — mention this to Claude directly

Step 6. Before you call it done

Run through this quick list with any new visual:

- It shows real data correctly when you load it with your fields in place
- Clicking or interacting with it behaves the way you expect
- It still looks right after resizing the dashboard or refreshing the data
- You've noted which settings (colors, labels, sizes) you can tweak yourself later

Remember: You can always return to the same Claude conversation later to ask for changes, add new features, or start a new visual using this one as a reference.

Saving as a Claude Skill

If you have a Claude Enterprise account, you can save all the instructions you provided as a Skill in Claude. This way it will only need to be set once, and Claude will use it as a reference for each new visual you create.

include in the Skill:

- Pyramid API links
- Required file structure
- Preferred CONFIG pattern
- Interaction rules
- Standard error handling expectations
- Preferred visual style guidance

Appendix A: Custom Visual API links

- Official reference:
<https://help.pyramidanalytics.com/2025/Content/Root/developer/reference/APIs/Custom%20Visual%20API/v2/Custom%20Visual%20APIs%202.0.htm>
- Aster Plot example (authoritative interaction pattern):
<https://help.pyramidanalytics.com/2025/Content/Root/developer/reference/APIs/Custom%20Visual%20API/v2/examples/Custom%20Visual%20Example%20-%20Aster%20Plot.htm>
- Datapoint API:
https://help.pyramidanalytics.com/Content/Root/developer/reference/APIs/Custom%20Visual%20API/v2/classes/api_s_objects.datapoint.html
- Canvas API:
https://help.pyramidanalytics.com/Content/Root/developer/reference/APIs/Custom%20Visual%20API/v2/classes/api_s_objects.canvas.html
- Utils API (selectDatapoints, inject, deselectAllElements):
https://help.pyramidanalytics.com/Content/Root/developer/reference/APIs/Custom%20Visual%20API/v2/classes/api_s_objects.utils.html
- Events Enum (Render, DataPointSelected, ElementSelected):
https://help.pyramidanalytics.com/Content/Root/developer/reference/APIs/Custom%20Visual%20API/v2/enums/api_s_objects.event.html