Figma Token Debugging

# Figma Token Debugging Guide

This document provides diagnostic methods for troubleshooting design tokens in Figma, particularly when working with Token Studio and variable detachment.

---

## Diagnostic Method: Finding Hidden Variables

### Problem

When trying to detach all variables from a Figma component, sometimes one variable remains attached but you can't identify which layer contains it or which variable it is.

### Solution: MCP Variable Detection

Use the Figma MCP (Model Context Protocol) tools to inspect the currently selected object and identify any remaining variables.

#### Step-by-Step Process

1. \*\*Select the problematic component\*\* in Figma
2. \*\*Use MCP variable detection\*\* to identify remaining variables:

mcp\_figma\_dev\_mod\_get\_variable\_defs

1. \*\*Analyze the results\*\* to identify:

* Variable name/path
* Variable value
* Layer where it's applied

#### Example Output

{

"ob/h/button/icon-only/surface/size/md": "36"

}

This tells you:

* \*\*Variable\*\*: `ob/h/button/icon-only/surface/size/md`
* \*\*Value\*\*: `36` (likely pixels)
* \*\*Type\*\*: Size token for icon-only button surface
* \*\*Layer\*\*: Applied to the surface/background layer of the component

#### Finding the Specific Layer

Based on the variable name, you can deduce:

* \*\*Token path\*\*: `ob/h/button/icon-only/surface/size/md`
* \*\*Component\*\*: Button
* \*\*Variant\*\*: Icon-only
* \*\*Property\*\*: Surface size
* \*\*Size\*\*: Medium (md)

**Most likely applied to:**

* Background/Surface layer
* Container layer
* Clickable area layer

#### Search Strategy

1. \*\*Layer names to check:\*\*

* "Background"
* "Surface"
* "Container"
* "Base"
* "Frame"

1. \*\*Properties to inspect:\*\*

* Width/Height (for size tokens)
* Padding (for spacing tokens)
* Border radius (for radius tokens)
* Colors (for color tokens)

1. \*\*Look for the exact value\*\* (36 in this example) in layer properties

---

## Common Variable Types and Their Layers

### Size Tokens

* \*\*Pattern\*\*: `ob/h/component/variant/surface/size/scale`
* \*\*Applied to\*\*: Background, container, or frame layers
* \*\*Properties\*\*: Width, height, or both

### Color Tokens

* \*\*Pattern\*\*: `ob/s/color/category/property/contrast/theme`
* \*\*Applied to\*\*: Fill, stroke, or text layers
* \*\*Properties\*\*: Fill color, stroke color, text color

### Spacing Tokens

* \*\*Pattern\*\*: `ob/s/spacing/category/scale`
* \*\*Applied to\*\*: Container layers, auto-layout frames
* \*\*Properties\*\*: Padding, gap, margins

### Typography Tokens

* \*\*Pattern\*\*: `ob/s/typography/category/property`
* \*\*Applied to\*\*: Text layers
* \*\*Properties\*\*: Font family, size, weight, line height

---

## Troubleshooting Workflow

### 1. Identify Remaining Variables

# Select component in Figma

# Run MCP command to get variable definitions

mcp\_figma\_dev\_mod\_get\_variable\_defs

### 2. Decode Variable Information

* Parse the token path to understand component/property
* Identify expected layer type from token category
* Determine likely property (size, color, spacing, etc.)

### 3. Locate the Layer

* Check layer panel for matching names
* Look for the specific value in layer properties
* Inspect nested layers and component variants

### 4. Verify and Detach

* Confirm the variable is applied to the identified layer
* Detach the variable manually
* Verify no other variables remain

---

## Prevention Strategies

### Component Setup

1. \*\*Use consistent layer naming\*\* (Background, Surface, Text, Icon)
2. \*\*Document variable applications\*\* in component descriptions
3. \*\*Test variable detachment\*\* during component development

### Token Management

1. \*\*Use semantic token names\*\* that clearly indicate purpose
2. \*\*Follow consistent naming conventions\*\* across all tokens
3. \*\*Document token-to-layer mappings\*\* for complex components

### Quality Assurance

1. \*\*Regular variable audits\*\* using MCP diagnostic tools
2. \*\*Test component detachment\*\* before publishing
3. \*\*Maintain variable usage documentation\*\*

---

## MCP Commands Reference

### Get Variable Definitions

mcp\_figma\_dev\_mod\_get\_variable\_defs

Returns all variables applied to the currently selected object.

### Get Component Image

mcp\_figma\_dev\_mod\_get\_image

Provides visual context for the selected component.

### Get Code Connect Mapping

mcp\_figma\_dev\_mod\_get\_code\_connect\_map

Shows component-to-code relationships if configured.

---

## Use Cases

### Scenario 1: Orphaned Size Token

* \*\*Problem\*\*: Can't detach size variable from button
* \*\*Diagnosis\*\*: MCP shows `ob/h/button/surface/size/md: 40`
* \*\*Solution\*\*: Check background layer width/height properties

### Scenario 2: Hidden Color Token

* \*\*Problem\*\*: Component still has color variable after detachment
* \*\*Diagnosis\*\*: MCP shows `ob/s/color/status/info/bg: #0066cc`
* \*\*Solution\*\*: Check nested layers, icon fills, or stroke colors

### Scenario 3: Typography Token Remnant

* \*\*Problem\*\*: Text appears to have variable but can't find it
* \*\*Diagnosis\*\*: MCP shows `ob/s/typography/body/size: 16`
* \*\*Solution\*\*: Check text layer font size or line height

---

## Best Practices

1. \*\*Always run diagnostics\*\* before declaring a component "clean"
2. \*\*Document findings\*\* for future reference
3. \*\*Use consistent layer naming\*\* to make diagnostics easier
4. \*\*Test with multiple component states\*\* (hover, disabled, etc.)
5. \*\*Create component variant matrices\*\* to catch edge cases

---

\*Last updated: July 10, 2025 - Created diagnostic method for Figma variable debugging\*