

Design Token Audit Playbook

Date	04.08.2026
Owner	Ivo Baldi
Purpose	Reusable guide for future design token audits

What This Playbook Is For

This playbook describes how to run a design token audit in a safe, traceable, and repeatable way.

It can be used for:

- new audits in existing Figma libraries
- follow-up audits after larger token changes
- baseline releases for Global, Android, or iOS libraries
- AI-supported token analysis
- handoffs between design, engineering, and design system governance

The goal is not only cleanup. The goal is a token library whose intent is clearly readable by people, engineering, and AI.

Core Principle

Every audit follows this pattern:

1. Read first.
2. Document findings with concrete numbers.
3. Plan small, prioritized cleanup passes.
4. Create a preview before every change.
5. Write only after explicit approval.
6. Verify after every change.
7. Document every relevant change in the changelog and with references.
8. Capture the final state as a baseline.

Before The Audit

Clarify Context

Before the first scan, clarify:

- Which Figma library is being audited?
- Is it a Global, Android, iOS, or another platform library?
- Is there an existing reference library?
- Are there existing changelog or reference structures?
- Who is the owner for changelog entries?
- Which areas are especially risky?
- Should the audit end with a new baseline version?

Gather Key Inputs

Before the audit, note:

- library name
- Figma file key
- relevant pages
- relevant collections
- expected modes
- current version or baseline
- owner / author for changelog entries
- known risks
- known open questions

Example:

```
Library: iOS Design Library
Figma file key: [file key]
Collections: Themes, Sizing
Expected modes: Light / Dark, default / ax
Owner: Ivo Baldi
Target: Token Audit Baseline v2.0.0
```

Read-only Scan

Every audit starts with a read-only scan. In this phase, no tokens, nodes, changelog rows, or references are changed.

Areas To Check

Check and document:

- local variable collections
- modes per collection
- variable counts per collection
- variable types, such as Color, Float, String, or Boolean
- empty descriptions
- existing description patterns
- scopes per collection
- tokens with `ALL_SCOPES`
- missing mode values
- Light/Dark values or default/AX values
- suspicious AX values, especially zero values
- Typography values and Typography scopes
- duplicate names
- leading or trailing spaces
- double spaces
- unwanted spaces around hyphens
- old or inconsistent naming patterns
- unresolved aliases
- local inbound aliases
- possible orphan candidates
- changelog structure
- changelog sort order
- author and type values in the changelog
- reference cards or reference frames

Read-only Report Format

A good read-only report includes:

```
Library:
File key:
Collections:
Total variables:
Modes:
Descriptions complete:
Empty descriptions:
Missing mode values:
ALL_SCOPES:
Naming issues:
Duplicate names:
Unresolved aliases:
Potential orphan candidates:
Changelog status:
Reference status:
Initial risk summary:
Recommended cleanup order:
```

Prioritized Cleanup Order

The recommended order is:

1. Mode cleanup
2. Scope cleanup
3. Changelog format alignment
4. Naming cleanup
5. Orphan review
6. Description passes
7. Baseline release

This order reduces risk because structural issues are made visible before many new descriptions or changelog entries are created.

Pass 1: Mode Cleanup

Goal:

Remove unwanted or outdated modes, but only when it is clear that no aliases, values, or consumer expectations will be affected.

Check:

- Which modes exist?
- Which modes are expected?
- Are all required modes populated?
- Are there aliases that depend on specific modes?
- Are Light/Dark or default/AX values correct?

Do not:

- Do not remove modes without checking impact first.
- Do not change AX modes if accessibility values could be affected.

Pass 2: Scope Cleanup

Goal:

Set scopes narrowly enough that tokens are only offered where they can be used meaningfully.

Check:

- Which tokens have `ALL_SCOPES` ?
- Which scopes make sense for Color tokens?
- Which scopes make sense for Float tokens?
- Are Typography tokens affected?
- Are there intentionally broad scopes?

Rule:

Non-Typography tokens can usually be cleaned first. Typography tokens must be handled cautiously and changed only after an explicit decision.

Pass 3: Changelog Format Alignment

Goal:

The library should have a consistent, traceable changelog structure.

Recommended format:

- Date
- Author
- Area
- Type
- Description
- Figma

Rules:

- newest entries appear at the top
 - Author is the real owner, not an AI agent
 - Type contains values such as Added, Changed, Fixed, Removed, Deprecated, Release, or Documented
 - dividers should match the reference library
 - if the reference library uses row bottom strokes, do not introduce separate divider rectangles
-

Pass 4: Naming Cleanup

Goal:

Token names should be consistent, machine-readable, and understandable for designers.

Check:

- duplicate names
- leading spaces
- trailing spaces
- double spaces
- inconsistent hyphen usage
- old naming patterns
- singular/plural inconsistencies
- platform patterns that do not fit the library

Rule:

Naming changes can be breaking changes. Before renaming, it must be clear whether consumer files or code are affected.

Pass 5: Orphan Review

Goal:

Classify unused tokens safely without accidentally deleting tokens that are still used in production.

Classification

Confirmed orphan:

- no local inbound aliases
- no canvas or style bindings
- no known consumer usage
- consumer usage has been sufficiently validated

Orphan candidate / hold:

- no local usage found
- consumer usage has not been fully checked
- the scan is incomplete, uncertain, or time-limited

Rule

Tokens must not be deleted only because they appear unused inside the library.

If consumer usage is not proven, candidates stay on hold.

Pass 6: Description Passes

Goal:

Every token should have a description that clearly explains intent and usage.

Recommended order:

1. Mapped tokens
2. Corner / radius tokens
3. Component color tokens
4. Component sizing tokens
5. Typography tokens

Rules:

- read existing descriptions first
- do not overwrite good existing descriptions
- only edit empty or explicitly approved descriptions
- keep descriptions short, clear, and usage-oriented
- do not repeat technical values if they are already visible in the token itself

Good description:

Used for the primary grouped background surface in standard app layouts.

Weak description:

Color token.

Pass 7: Baseline Release

Goal:

Document the achieved state as a clear, findable reference point.

A baseline should include:

- visible baseline marker in the library
- final changelog row
- reference card or reference frame
- final verification report
- separate documentation for hold items
- clear next-step recommendation

Example:

Token Audit Baseline v2.0.0

Preview Before Every Change

Every Figma change needs a concrete preview before it is made.

The preview must include:

- affected library
- affected collection
- type of change
- exact number of affected tokens or nodes
- affected names or naming patterns
- risk level
- planned changelog entry
- planned reference card or reference frame
- verification plan

Example:

Preview: Scope Cleanup

Library: iOS Design Library

Collection: Themes

Affected tokens: 24

Change: Remove ALL_SCOPES from non-Typography color tokens

Risk: Low to medium

Changelog: 07.07.2026 / Ivo Baldi / Tokens / Fixed / Cleaned broad scopes for non-Typography theme tokens

Reference: Scope cleanup reference card in References / Design System Changelog

Verification: Re-scan Themes scopes and confirm 0 non-Typography ALL_SCOPES

Approval Rule

Approval applies only to the specific pass described in the preview.

If a preview for scope cleanup is approved, that approval does not automatically apply to:

- naming cleanup
- description updates
- orphan deletion
- Typography changes
- AX changes
- changelog restructuring

Every pass needs its own preview and approval.

Verification After Every Change

After every approved pass, check again:

- Was exactly the planned number of items changed?
 - Were the target issues resolved?
 - Were there any unexpected changes?
 - Are modes still complete?
 - Were descriptions preserved or correctly added?
 - Are scopes still meaningful?
 - Are changelog entries and references present?
 - Are reference cards visible and readable?
-

Final Baseline Checklist

An audit is only complete once these points have been checked:

- variable totals documented
- collection totals documented
- expected modes present
- empty descriptions are 0 or intentionally documented
- missing mode values are 0
- unresolved aliases are 0
- naming issues are 0 or intentionally documented
- duplicate names are 0
- unwanted `ALL_SCOPES` have been cleaned
- Typography exceptions are documented
- AX values have been checked
- suspicious AX zero values are 0 or explained
- changelog exists
- changelog is sorted newest first
- author values are correct
- type values are correct

- reference cards or frames are visible
 - hold items are documented separately
 - next step is clear
-

What Should Never Be Changed Automatically

These areas need special care:

- Typography values
- AX or accessibility values
- tokens with possible consumer usage
- alias structures with platform logic
- direct alpha values
- intentionally documented special cases
- names of production-used tokens
- modes that may be expected in consumer files

Rule:

If the impact is not clearly understood, document instead of changing.

Documentation Outputs

After an audit, at least these artifacts should exist:

- audit handoff
- release note
- changelog entry in Figma
- reference card or reference frame in Figma
- list of hold items
- final baseline numbers

Optional but useful:

- Teams or Slack update
- portfolio or case study text
- consumer usage audit for orphan candidates
- follow-up audit plan

Template: Audit Handoff

```
# [Library Name] Token Audit Handoff [Version]
```

```
Date:
```

```
Owner:
```

```
Library:
```

```
Figma file key:
```

```
## Status
```

```
[Baseline status and short summary.]
```

```
## Final Verification
```

- Local variables:
- Collections:
- Modes:
- Descriptions:
- Empty descriptions:
- Duplicate variable names:
- Naming issues:
- Missing mode values:
- Unresolved aliases:
- ALL_SCOPES:

```
## Completed Audit Work
```

- [Completed pass]
- [Completed pass]
- [Completed pass]

```
## Changelog / References
```

- Changelog rows:
- Reference cards:
- Author values:
- Type values:
- Sort order:
- Visibility:

```
## Hold / Follow-up
```

- [Hold item]
- [Reason]
- [Recommended next step]

Recommended Next Step

[Clear recommendation.]

Template: Release Note

[Library Name] Token Audit Release Note [Version]

Date:

Owner:

Library:

Summary

[Short summary of what changed and why it matters.]

Verified State

- [Metric]
- [Metric]
- [Metric]

Completed

- [Completed item]
- [Completed item]

On Hold

[Hold items and reason.]

Next Step

[Clear next action or state that no further cleanup is required.]

Template: AI Prompt For Future Audits

Use the Design Token Audit Playbook.

Start read-only. Do not change anything in Figma until you have shown a concrete preview and I explicitly approve the specific pass.

Audit this library:

Library:

Figma file key:

Relevant pages:

Collections:

Expected modes:

Owner for changelog:

Target baseline:

First, report:

- collections and modes
- variable counts by collection and type
- empty descriptions
- missing mode values
- scopes and ALL_SCOPES
- naming issues
- unresolved aliases
- orphan candidates
- changelog and reference structure
- recommended pass order

Treat Typography and AX values as high risk. Do not change them unless I explicitly approve that exact change.

Existing Baseline References

Global Tokens:

- Baseline: v1.0.0
- Variables: 265

- Descriptions: 265/265 complete
- ALL_SCOPES: 0
- Unresolved aliases: 0

iOS Design Library:

- Baseline: v2.0.0
 - Variables: 375
 - Descriptions: 375/375 complete
 - Themes ALL_SCOPES: 0
 - Sizing ALL_SCOPES: 43, Typography only
 - Orphan candidates on hold: 85
-

Short Version

A good design token audit is not one large cleanup. It is a controlled workflow:

Read, understand, preview, approve, change, verify, document.

This keeps the token library not only clean, but trustworthy for future design system work and AI-supported workflows.