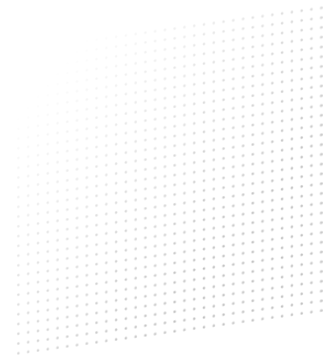


Semantic token naming.



A closed grammar and the words allowed in it.

One shape per category. One word per concept.

22

rules, 16 of them enforced by the linter

30

bad names in the fixture, all rejected

11

of 22 legacy names that needed a person

The semantic layer is an API. Its names are the API.

Application code, and every agent writing it, reaches the design system through semantic token names. A name like `color.bg.brand-hover` is a promise: this is the hovered fill of a primary action, in every theme. When names are invented one pull request at a time, the promise breaks in small ways. `background` and `bg` both appear, `error` and `danger` mean the same thing, `primary` means three things, and an agent guessing `color.text.secondary` gets a name that does not exist.

This guide closes the grammar. Each category has one shape (`color.<property>.<variant>`, `space.<axis>.<scale>`), each slot takes words from a controlled vocabulary, synonyms map to one preferred term and ambiguous words become decisions for a person. The kit makes it executable: `naming.vocabulary.json` holds the grammar and the words, `lint-token-names.mjs` enforces 16 of the 22 rules and `rename-map.mjs` turns a legacy set into a migration map.

The linter accepts all 23 semantic tokens from Field Guide 03's token files, with one warning for the deprecated `color.link`. Field Guide 22 draws the same tokens as a diagram.

Version 1.0 / Sources checked 24 September 2026

Field Guide 23 of the Design × AI series. Pairs with Field Guide 03 (token layer rules, whose rule R06 this guide expands) and 22 (token architecture diagram). Verified 24 September 2026 with Node 22.22 against DTCG 2025.10.

Practical guidance, not a standard. The vocabulary is a starting set drawn from the systems compared on the next page; change the words, keep the rules. The essay this guide belongs to printed an earlier version of the convention (`color.background.*`, `color.interactive.*`); Appendix C migrates it. Prepared with AI assistance and edited by hand.

START HERE

Every system names the same decisions differently.

Nathan Curtis describes token names as levels: namespace, object, base (category, concept, property) and modifiers (variant, state, scale, mode). Mature systems use the same levels with different words. Pick yours once, then enforce them.

	This guide	Atlassian	Primer	Material 3	Spectrum
Background	<code>color.bg.*</code>	<code>color.background.*</code>	<code>bgColor-*</code>	<code>surface</code> , <code>*-container</code>	<code>*-background-color-*</code>
Foreground	<code>color.text.*</code> , <code>on-*</code>	<code>color.text.*</code>	<code>fgColor-*</code>	<code>on-*</code>	<code>*-content-color-*</code>
States	<code>-hover</code> , <code>-active</code>	<code>.hovered</code> , <code>.pressed</code>	<code>-rest</code> , <code>-hover</code>	none in role names	<code>-hover</code> , <code>-down</code>
Example	<code>color.bg.brand-hover</code>	<code>...bold.hovered</code>	<code>button-primary-bgColor-hover</code>	<code>on-primary</code>	<code>accent-background-color-hover</code>

From each system's published token names (S03 to S07). Salesforce SLDS 2 global hooks follow the same idea with `--slds-g-color-surface-1` and `--slds-g-color-on-surface-2` (S08).

Starting from zero

Sections 01 and 02. Copy `naming.vocabulary.json`, delete the categories you do not need, and add words only when a real decision needs one.

Cleaning up a live set

Run the linter in warning mode, then `rename-map.mjs` on the failures. Section 04 and Appendix C.

Opening it to agents

Item N21: give agents the grammar and the linter as a verify step, so a guessed name fails fast.

Adding a word

Item N17. A new word is a reviewed change to the vocabulary file, not a side effect of a new token.

Label	Meaning
CHECK	Enforced by <code>lint-token-names.mjs</code> ; the rule code in the output matches the item id.
DTCG	Required by the DTCG 2025.10 Format module.
SCRIPT	Done or checked by <code>rename-map.mjs</code> .
AGENT	Exists so an agent can find and use the right name.
PRACTICE	A working method with a review signal rather than a hard gate.

SECTION 01

Shape: one grammar per category.

A closed grammar means every valid name matches a template, and a name that matches none is a design question before it is a pull request.

Suggested owners: Design-system lead + token owner

☐ Segments are lowercase kebab-case ASCII

N01

Dots separate levels; hyphens join words inside a level. No capitals, underscores or spaces, so every platform transform starts from the same string.

Evidence: `color.Text.default` and `color.text.default_hover` fail N01.

DTCG names are case-sensitive, so `Text` and `text` would be two tokens. S01.

☐ No DTCG reserved characters

N02

A name never starts with `$` and never contains `{`, `}` or a `.` inside a key. Braces are alias syntax; the dot is the group separator.

Evidence: `color.text.{brand}`, `$color.text.default` and the key `"muted.v2"` fail N02.

DTCG 2025.10 Format: forbidden characters in token and group names. S01.

☐ The first segment is a category from a closed list

N03

`color`, `space`, `size`, `radius`, `stroke`, `font`, `typography`, `shadow`, `z`, `motion`, `opacity`. The system prefix (`--ds-`) is added by the build, not written into the path, and `core.` or `comp.` names belong to other layers.

Evidence: `brand.primary.fill` and `core.color.blue.600` fail N03.

Curtis: category is the base level; namespaces are prepended by the system. S02.

☐ Each category has a fixed depth

N04

`color.<property>.<variant>` is three segments, `radius.<object>` is two. A state is part of the variant, not a fourth level, so names sort and diff together.

Evidence: `color.text` and `color.bg.brand.hover` fail N04.

Kit grammar; 13 templates across 11 categories.

☐ Inside the variant, order is fixed

N05

`[on-]role[-state]`: `on-` first, the role, a state last. `disabled` takes no state.

Evidence: `color.bg.hover-brand` and `color.text.muted-on-brand` fail N05.

Curtis: modifiers are appended last. S02.

☐ No two names collide after a platform transform

N06

`color.bg.brand-hover` and `color.bg-brand.hover` both become `--color-bg-brand-hover` and `colorBgBrandHover`. The linter transforms every name and fails on duplicates.

Evidence: The fixture's collision group of three names fails N06.

DTCG notes that case-only differences can collide in code generation. S01.

If a name fits no template, the template is the discussion.

SECTION 02

Words: a controlled vocabulary.

The grammar says where words go. The vocabulary says which words exist. Both live in one JSON file that the linter, the rename generator and your agents read.

Suggested owners: Design-system lead + content designer

☐ **Every slot takes words from its list**

N07 **CHECK**

Roles are per property: `link` is a text role, not a border role; `surface` is a background role, not a text role. The kit's vocabulary has 52 slot words and 17 color roles.

Evidence: `color.text.fancy`, `color.border.link` and `space.inline.huge` fail N07.

Curtis: form a controlled vocabulary with preferred terms. S02.

☐ **One word per concept; synonyms name the preferred term**

N08 **CHECK**

`background` is `bg`, `error` is `danger`, `medium` is `md`, `pressed` is `active`. Some words are right for one property only: for text, `subtle` is `muted`. Ambiguous words (`primary`, `secondary`, `accent`, `interactive`) are rejected with the choices listed.

Evidence: Six fixture names break N08 first; each error prints the preferred term or the choices.

Curtis: avoid homonyms; `primary` aka `default`, `secondary` aka `subdued` or `subtle`. S02.

☐ **No values in semantic names**

N09 **CHECK**

No hues (`blue`, `gray`), palette steps (`100`) or units (`16`, `px`). A value in the name stops being true the first time the value changes.

Evidence: `color.text.blue`, `space.inline.16` and `color.bg.gray-100` fail N09.

Field Guide 03 R02 is the mirror rule: core names describe values, semantic names do not.

☐ **No mode words**

N10 **CHECK**

`light`, `dark`, `hc`, `dim` never appear. Modes swap values behind the same name through the resolver.

Evidence: `color.text.default-dark` fails N10.

Field Guide 03 R08; DTCG resolver contexts.

☐ **No component or page names**

N11 **CHECK**

`button`, `card`, `checkout`. A decision that belongs to one component is a `comp.<name>.*` token, created under Field Guide 03 R12. Stacking layers (`z.modal`, `z.tooltip`) are the one listed exception.

Evidence: `color.bg.button` fails N11 with the hint `comp.button.*`.

Curtis: start within a component, promote across components later. S02.

☐ **on- names a foreground on a filled background**

N12 **CHECK**

`on-*` is for text and icons only, and only for roles that are fills: `brand`, `danger`, `warning`, `success`, `info`, `inverse`. It is the pairing convention Material 3 and SLDS 2 use.

Evidence: `color.bg.on-brand` and `color.text.on-link` fail N12.

Material 3 `on-primary`, `on-surface`; SLDS 2 `on-surface`. S06, S08.

A synonym is a second API for the same decision.

SECTION 03

Meaning: names that keep their promise.

A name can be well formed and still lie: a foreground with no background to sit on, a hover state with no resting state, a token nobody can explain. These rules look across the whole set.

Suggested owners: Design-system lead + accessibility lead

☐ **Pairs are complete**

N13 **CHECK**

Every `color.text.on-*` has the `color.bg.*` it names to sit on, so the contrast pair Field Guide 03 checks can exist.

Evidence: `color.text.on-success` without `color.bg.success` fails N13.

Field Guide 03 R09 checks each pair's contrast.

☐ **Every state has a resting base**

N14 **CHECK**

`color.bg.brand-hover` requires `color.bg.brand`. A hover without a rest is a name for a value nobody sees first.

Evidence: `color.border.focus-hover` alone fails N14.

Kit rule.

☐ **Every name says when to use it**

N15 **CHECK**

A `$description` on every semantic token. The name says what it is for in two words; the description says when, in one sentence.

Evidence: A token without `$description` fails N15 when the linter reads token files.

Field Guide 03 R10.

☐ **Legacy names stay only while deprecated, with a valid replacement**

N16 **CHECK** **DTCG**

A name outside the grammar may remain for one major version if `$deprecated` names a replacement that exists and passes. Then it is a warning, not an error.

Evidence: Field Guide 03's `color.link` passes with one warning; `$deprecated: true` and a missing replacement fail N16.

DTCG `$deprecated`; Field Guide 03 R22. S01.

A name is a promise the whole set has to keep.

SECTION 04

Change: words and renames are data.

The vocabulary grows, and names get fixed. Both happen through files a machine can check, so the next person and the next agent see the same decision.

Suggested owners: Design-system lead + release owner

☐ **The vocabulary is versioned, and a new word is reviewed**

N17 **PRACTICE**

`naming.vocabulary.json` has a version and an owner. Adding a role, a scale step or a category is a reviewed change with a reason, not a side effect of a token pull request.

Evidence: CODEOWNERS on the vocabulary file; its version bumps in the same change.

Curtis: do not globalize decisions prematurely. S02.

☐ **Renames ship as a map**

N18 **SCRIPT**

`rename-map.mjs` writes `token-renames.json` in the Field Guide 03 R24 format, so a codemod or an agent applies renames mechanically.

Evidence: The legacy fixture produces a 16-entry map with version 3.0.0.

Field Guide 03 R24.

☐ **The map is checked before it ships**

N19 **SCRIPT**

Every target passes the linter, chains collapse (a to b to c becomes a to c), two names may land on one target only as a declared merge, and the migrated set passes N06, N13 and N14.

Evidence: Removing the declared merge for `color.bg.subtle` from the overrides fails the run.

Kit script.

☐ **Ambiguous names become decisions, never guesses**

N20 **SCRIPT**

The generator renames only what the synonym tables settle. Anything ambiguous or still invalid is listed for a person, whose answer and reason go into an overrides file.

Evidence: Of the essay's 22 legacy names, 5 rename mechanically, 6 stay and 11 need a decision.

Kit script; Appendix C.

☐ **Agents get the grammar and the linter**

N21 **AGENT**

Examples teach an agent the style; the grammar and a failing linter teach it the rules. The kit's instruction block tells agents to use existing tokens, never invent one and run the linter before finishing.

Evidence: `agents/AGENTS.naming.md` is in the agent's context and names the verify command.

Field Guide 04 covers where agent instructions live.

☐ **The linter runs where token files change**

N22 **PRACTICE**

Pre-commit and CI run `lint-token-names.mjs` on changed token files next to Field Guide 03's rules check and Field Guide 22's diagram check. A non-zero exit blocks the merge.

Evidence: One `names:lint` script in `package.json`, used in both places.

Kit `package.json`.

A rename nobody wrote down is a breaking change nobody announced.

APPENDIX A

The grammar and the words.

The complete set from `naming.vocabulary.json` version 1.0.0. Edit the words for your system; the linter reads whatever the file says.

Template	Slot words
<code>color.{property}.{variant}</code>	property: <code>text</code> , <code>icon</code> , <code>bg</code> , <code>border</code> . variant: <code>[on-]role[-state]</code> , <code>states</code> <code>hover</code> , <code>active</code>
roles for <code>text</code>	<code>default</code> , <code>muted</code> , <code>brand</code> , <code>danger</code> , <code>warning</code> , <code>success</code> , <code>info</code> , <code>link</code> , <code>disabled</code> , <code>selected</code>
roles for <code>icon</code>	as text, without <code>link</code>
roles for <code>bg</code>	<code>surface</code> , <code>subtle</code> , <code>raised</code> , <code>overlay</code> , <code>inverse</code> , <code>brand</code> , <code>danger</code> , <code>warning</code> , <code>success</code> , <code>info</code> , <code>disabled</code> , <code>selected</code>
roles for <code>border</code>	<code>default</code> , <code>subtle</code> , <code>strong</code> , <code>focus</code> , <code>brand</code> , <code>danger</code> , <code>warning</code> , <code>success</code> , <code>info</code> , <code>disabled</code> , <code>selected</code>
<code>space.{axis}.{scale}</code>	axis: <code>inline</code> , <code>stack</code> , <code>inset</code> . scale: <code>xs</code> , <code>sm</code> , <code>md</code> , <code>lg</code> , <code>xl</code>
<code>size.{object}.{scale}</code>	object: <code>control</code> , <code>icon</code>
<code>radius.{object}</code> , <code>stroke.{purpose}</code>	<code>control</code> , <code>container</code> , <code>full</code> ; <code>default</code> , <code>focus</code>
<code>font.family.*</code> , <code>font.weight.*</code>	<code>body</code> , <code>heading</code> , <code>code</code> ; <code>regular</code> , <code>medium</code> , <code>semibold</code> , <code>bold</code>
<code>typography.{role}.{scale}</code>	<code>display</code> , <code>heading</code> , <code>body</code> , <code>label</code> , <code>code</code> ; <code>sm</code> , <code>md</code> , <code>lg</code>
<code>shadow.*</code> , <code>z.*</code>	<code>raised</code> , <code>overlay</code> ; <code>base</code> , <code>dropdown</code> , <code>sticky</code> , <code>overlay</code> , <code>modal</code> , <code>toast</code> , <code>tooltip</code>
<code>motion.duration.*</code> , <code>motion.easing.*</code> , <code>opacity.*</code>	<code>instant</code> , <code>fast</code> , <code>moderate</code> , <code>slow</code> ; <code>standard</code> , <code>enter</code> , <code>exit</code> ; <code>disabled</code> , <code>scrim</code>

Abbreviations allowed: `bg`, `xs`, `sm`, `md`, `lg`, `xl`, `z`. Everything else is spelled out. Curtis suggests a proportional scale rather than t-shirt sizes for generic space; this vocabulary keeps t-shirt sizes because semantic space tokens name a purpose (control padding), matching Field Guide 03.

`naming.vocabulary.json` (excerpt)

```

"synonyms": {
  "background": "bg", "error": "danger", "pressed": "active",
  "medium": "md", "spacing": "space", "colour": "color"
},
"propertySynonyms": {
  "text": { "subtle": "muted", "secondary": "muted", "inverse": "on-inverse" },
  "bg": { "muted": "subtle", "default": "surface" }
},
"ambiguous": {
  "primary": "brand for fills and borders, default for text, surface for page backgrounds",
  "interactive": "a role (brand, link) plus a state"
}

```

25 synonyms and 8 ambiguous words in the full file. The same tables drive the linter's messages and the rename generator.

APPENDIX B

What the linter prints.

Real output from the kit. The ZIP includes Field Guide 03's semantic token files under

`tokens/semantic/`.

terminal: Field Guide 03's semantic tokens

```
$ node scripts/lint-token-names.mjs tokens/semantic/light.tokens.json tokens/semantic/dark.tokens.json
warning N16 color.link: legacy name kept until removal; replaced by color.text.link

23 names checked: 0 rejected, 0 errors, 1 warnings.
```

terminal: the bad-names fixture (excerpt)

```
$ node scripts/lint-token-names.mjs --names examples/bad-names.json
error  N02 color.text.{brand}: "{brand}" uses a character DTCG reserves ({, } or a leading $)
error  N05 color.bg.hover-brand: the state "hover" comes last: brand-hover
error  N07 color.border.link: "link" is not a border role; border takes default, subtle, strong, focus, brand,
danger, warning, success, info, disabled, selected
error  N08 color.bg.primary: "primary" is ambiguous; decide: brand for fills and borders, default for text, surface
for page backgrounds
error  N08 color.text.subtle: for text, "subtle" is "muted"
error  N09 color.bg.gray-100: "gray" describes the value, not the use; hues belong in core
error  N11 color.bg.button: "button" is a component or page; component-specific tokens live in comp.button.*
error  N12 color.bg.on-brand: on-* names a foreground; use it on text or icon, not bg
error  N06 color.bg-brand.hover: collides with color.bg.brand.hover, color.bg.brand-hover as --color-bg-brand-hover
and colorBgBrandHover
error  N13 color.text.on-success: pairs with color.bg.success, which does not exist

30 names checked: 30 rejected, 35 errors, 0 warnings.
```

Each fixture entry records the rule it breaks first; the kit's test checks that every name fails with that rule. Some names break two rules, hence 35 errors for 30 names.

terminal: file-level rules

```
$ node scripts/lint-token-names.mjs examples/bad.tokens.json
error  N02 color.text.muted.v2: key "muted.v2" in bad.tokens.json uses {, } or . which DTCG forbids in names
error  N15 color.text.default: no $description: say when to use it
error  N16 color.text.caption: deprecated without naming its replacement in $deprecated
error  N16 color.text.legacy-link: replacement color.text.hyperlink does not exist
error  N04 color.text.muted.v2: color names have the shape color.{colorProperty}.{colorVariant}

4 names checked: 4 rejected, 5 errors, 0 warnings.
```

APPENDIX C

Migrating the essay's first convention.

The essay printed a first version of this convention: `color.background.*`, `color.interactive.*`, `spacing.component.*`. `examples/legacy-names.json` holds those 22 names. The generator settles half; a person settles the rest.

terminal: first pass, no decisions yet (excerpt)

```
$ node scripts/rename-map.mjs examples/legacy-names.json
rename    color.background.muted -> color.bg.subtle
rename    color.text.inverse -> color.text.on-inverse
rename    color.text.error -> color.text.danger
decide    color.background.primary: ambiguous: "primary" (brand for fills and borders, default for text, surface for
page backgrounds)
decide    color.interactive.hover: ambiguous: "interactive" (a role (brand, link) plus a state)
decide    spacing.layout.section: no mechanical rename: space.layout.section still breaks N07 ("layout" is not a
spaceProperty word; use inline, stack, inset)

22 names: 5 renamed, 6 unchanged, 11 need a decision, 0 problems.
```

Exit code 1 until every decision is made.

examples/rename-overrides.json (excerpt)

```
"decisions": {
  "color.background.primary": { "to": "color.bg.surface", "reason": "Page and card background." },
  "color.background.secondary": { "to": "color.bg.subtle", "reason": "Quiet fill; one palette step from muted." },
  "color.interactive.hover": { "to": "color.bg.brand-hover", "reason": "Hover state of the primary fill." }
},
"merges": {
  "color.bg.subtle": "secondary and muted backgrounds were one palette step apart; merged on purpose."
}
```

Example decisions for the fixture. Make yours from real usage and keep the reason.

examples/token-renames.json (generated, excerpt)

```
{
  "version": "3.0.0",
  "renames": {
    "color.background.accent": "color.bg.selected",
    "color.background.muted": "color.bg.subtle",
    "color.interactive.default": "color.bg.brand",
    "spacing.component.padding": "space.inset.md"
  }
}
```

Second pass with the overrides: 16 renamed, 6 unchanged, 0 decisions, 0 problems. Same shape as Field Guide 03's `token-renames.json`.

KEEP WITH THE VOCABULARY

Leave a vocabulary change record.

A record for one change to the grammar or the words. It is the trail that explains why a word exists, so nobody adds its synonym next year.

Vocabulary version / date	Words added / removed / moved
The decision the new word names	Synonyms and ambiguous words updated
Linter run on all token files (link)	Rename map generated (link), if names change
Decisions made, with reasons	Agent instruction block updated
Reviewer / owner	Next review trigger

Never add a word to make one token pass.
If a single new token needs a new word, the question is whether the decision exists, not whether the linter can be quieted.

SOURCES / MAINTENANCE

Keep the guide current.

Sources checked 24 September 2026. Token names quoted from each system's published tokens on that date. The linter and the rename generator were run on the kit's fixtures and on Field Guide 03's token files.

S01 / DTCG, Format module 2025.10

Names must not begin with \$ or contain {, } or .; names are case-sensitive; \$deprecated.

<https://www.designtokens.org/TR/2025.10/format/>

S02 / Nathan Curtis, Naming Tokens in Design Systems

Levels (namespace, object, base, modifier), controlled vocabulary, homonyms, order, promotion from local to global (EightShapes, October 2020).

<https://medium.com/eightshapes-llc/naming-tokens-in-design-systems-9e86c7444676>

S03 / Atlassian, design tokens

Foundation, property and modifier; color.icon.success as the worked example.

<https://atlassian.design/foundations/tokens/design-tokens>

S04 / Atlassian, all design tokens

Emphasis and state words: subtlest, subtler, subtle, bolder, hovered, pressed.

<https://atlassian.design/components/tokens/all-tokens>

S05 / GitHub Primer, token names

Base, functional and component tokens; bgColor, fgColor, borderColor; rest and hover.

<https://primer.style/product/primitives/token-names/>

S06 / Material Web, md-sys-color tokens

Material 3 color roles as shipped: primary, on-primary, primary-container, on-surface, outline.

https://github.com/material-components/material-web/blob/main/tokens/versions/v0_192/_md-sys-color.scss

S07 / Adobe Spectrum, color alias tokens

accent-background-color-hover, accent-content-color-default, down and key-focus states, disabled-background-color.

<https://github.com/adobe/spectrum-design-data/blob/main/packages/tokens/src/color-aliases.json>

S08 / Salesforce, global styling hooks

SLDS 2 hooks: --slds-g-color-surface-1, --slds-g-color-on-surface-2, --slds-g-spacing-3.

<https://www.salesforce.com/blog/what-are-styling-hooks/>

Maintenance: review the vocabulary when a new category or role is proposed, and at each DTCG release for changes to naming rules. Recheck the comparison table against the five systems once a year. Update the PDF, HTML, Markdown and JSON together.