ZEEK INFO PEEK

ZIP Functionality Inventory and Documentation Workflow

ZIP Documentation Suite • Spectrum 2 branded export • Generated 2026-02-11 13:50:59

ZIP Functionality Inventory and Documentation Workflow

This document explains how to identify, classify, and document all existing and new ZIP functionality for user-facing releases.

Quick Start (Urgent Line Paged)

For immediate operations before full documentation review:

1. Install ZIP in Dev Mode from `/Users/minnick/Documents/PASS/ZIP/zip-chrome-extension`.
2. Open Zendesk dashboard.
3. Open ZIP side panel and authenticate.
4. Use `Assigned Tickets` for immediate triage.
5. Return to this workflow doc for full release evidence mapping.

1. Documentation Objective

Keep user training aligned with reality by treating source code as the canonical source of truth.

2. Canonical Source Map

Use these files as the system map:

- `manifest.json`: permissions, commands, action behavior, version.
- `background.js`: Side Panel lifecycle, context menus, commands, tab scoping.
- `content.js`: Zendesk API calls, normalization, status filtering, auth error handling.
- `sidepanel.html`: user-visible controls and section structure.
- `sidepanel.js`: UX behavior, event handlers, workflows, exports, status text.
- `sidepanel.css`: interaction constraints and layout behavior.
- `README.md`: current user guidance baseline.

3. Functionality Inventory Template

For each feature, capture this schema:

Feature ID	User-facing Name	Primary User	Trigger	Preconditions	Step Summary	Output	Failure Mode	Source Files

Use IDs like `ZIP-OPEN-001`, `ZIP-TICKETS-005`, `ZIP-API-009`.

4. Current ZIP Feature Inventory (v1.0.21)

Feature ID	User-facing Name	Trigger	Output	Source Files
ZIP-OPEN-001	Open panel by extension icon	Action click	ZIP side panel opens	`manifest.json`, `background.js`
ZIP-OPEN-002	Open panel via context menu	Context menu item	Side panel opens	`background.js`
ZIP-OPEN-003	Toggle panel via keyboard	`Ctrl+Shift+Y` / `Command+Shift+Y`	Open/close panel (where API supports close)	`manifest.json`, `background.js`
ZIP-OPEN-004	Open horizontal workspace	Context menu/command	Full tab `sidepanel.html?mode=workspace`	`background.js`, `sidepanel.css`, `sidepanel.js`
ZIP-SCOPE-001	Domain-aware availability	Tab URL change	Enabled only on Zendesk origin	`background.js`
ZIP-AUTH-001	Login with Zendesk	Login button	Opens Zendesk sign-in URL	`sidepanel.html`, `sidepanel.js`
ZIP-AUTH-002	Session auto-detect polling	Logged-out idle	Auto-refresh when login detected	`sidepanel.js`
ZIP-AUTH-003	Sign out	Sign out button	Zendesk logout flow	`sidepanel.js`, `content.js`
ZIP-TICKETS-001	Assigned tickets	Assigned Tickets link/default load	Active tickets table	`sidepanel.js`, `content.js`
ZIP-TICKETS-002	Filter by Group or Group member	By Group selector	Group/team scoped table	`sidepanel.js`, `content.js`
ZIP-TICKETS-003	Filter by View	By View selector	View-scoped table	`sidepanel.js`, `content.js`
ZIP-TICKETS-004	Filter by Org	By Org selector	Org-scoped table	`sidepanel.js`, `content.js`

Feature ID	User-facing Name	Trigger	Output	Source Files
ZIP-TICKETS-005	Search + status filter	Search input + status dropdown	Refined visible rows	`sidepanel.js`
ZIP-TICKETS-006	Sort columns	Ticket header click	Asc/desc sort	`sidepanel.js`
ZIP-TICKETS-007	Open ticket in Zendesk	Row click or ticket ID click	Main tab navigates to ticket URL	`sidepanel.js`
ZIP-TICKETS-008	Refresh active source	Refresh icon	Reload same source context	`sidepanel.js`
ZIP-EXPORT-001	Export visible table to CSV	CSV button	Context-aware CSV download	`sidepanel.js`
ZIP-API-001	Zendesk GET runner	API path + params + Run GET	GET response rendered	`sidepanel.js`, `paths.js`, `content.js`
ZIP-API-002	Raw JSON download	Raw.JSON link	JSON file download	`sidepanel.js`
ZIP-UX-001	Side panel side awareness	getLayout context	Left/right visual adaptation	`background.js`, `sidepanel.js`, `sidepanel.css`
ZIP-UX-002	Narrow-width menu containment	Resize panel thinner	Dropdowns remain inside shell	`sidepanel.css`
ZIP-BRAND-001	Master icon system	Extension install/action icon	Unified ZEEK INFO PEEK iconography	`manifest.json`, `icons/`, `assets/brand/icons`
ZIP-BRAND-002	Branded login shell	Logged-out view	Splash + mark + mission-aligned shell	`sidepanel.html`, `sidepanel.css`, `assets/brand/splash`, `assets/brand/source`

Feature ID	User-facing Name	Trigger	Output	Source Files
ZIP-BRAND-003	Branded docs exports	Training doc generation	Spectrum 2 PDFs aligned to master kit	`docs/*.md`, `docs/assets/brand`, `docs/html`, `docs/pdf`

5. Classifying Existing vs New for Release Notes

Use this rule set each release:

- Existing: Feature present in prior tagged version and behavior unchanged.
- Changed: Feature existed, but output, trigger, text, or constraints changed.
- New: Feature not available in prior tagged version.
- Retired: Feature removed from user access.

6. Repeatable Change Detection Workflow

1. Pick baseline and target:

- Baseline: last release tag or prior commit.
- Target: current release commit.

2. Run targeted diff:

- `git diff <baseline>..<target> -- zip-chrome-extension/manifest.json zip-chrome-extension/background.js zip-chrome-extension/sidepanel.html zip-chrome-extension/sidepanel.js zip-chrome-extension/content.js zip-chrome-extension/sidepanel.css zip-chrome-extension/README.md`

3. Map each diff hunk to feature IDs.
4. Update inventory rows with `New/Changed/Existing/Retired` status.
5. Update training guide and UAT cases if feature behavior changed.

7. User-Facing Release Notes Template

Release `<version>` - ZIP

- Mission impact summary in one sentence.
- New:
- Changed:
- Fixed:
- Known limitations:
- User action required (if any):

8. Evidence Checklist Before Publishing User Docs

- Version number matches `manifest.json`.
- Every user-visible button/menu in `sidepanel.html` has documentation.
- Every command/context-menu entry has documentation.
- Every download behavior (CSV/JSON) has expected result text.
- UAT pass recorded for launch triggers and ticket workflows.

9. Documentation Ownership Model

- Product owner: approves wording for mission/value alignment.
- Technical owner: validates behavior against source.
- Support owner: validates troubleshooting clarity.
- Training owner: validates onboarding flow for new users.

10. Minimum Artifacts Per Release

- PRINT THIS FIRST quick-start card PDF for urgent onboarding.
- Management release memo with quick-start-first install path.
- Updated training guide.
- Updated functionality inventory.
- Updated UAT test sheet with pass/fail evidence.
- One-page release note for users.
- Regenerated branded HTML/PDF package via `scripts/render_branded_docs.sh`.

PASS • ZIP (Zeek Info Peek) Master brand kit aligned