Handz V1 PRD Bundle — Chapter CH32

Accessibility & Localization Readiness

Doc ID	HZ-V1-CH32_Accessibility_And_Localization_Readiness_R1
Revision	R1 (2026-01-02)
Status	Draft
Depends on	CH02, CH04, CH05, CH06, CH08, CH12, CH20, CH21, CH22, CH25, CH31
Related	CH33, CH35, CH36, CH37, CH38
Supersedes	—
Owned Decisions	Accessibility rules & acceptance tests; VoiceOver + Dynamic Type specs; localization readiness standards; string key conventions; RTL handling policy; a11y QA workflow.
Open Questions / Placeholders	See §12 Placeholder Registry (CH32-owned).

1. Purpose and scope

This chapter defines the minimum accessibility (a11y) and localization-readiness requirements for Handz V1 (iOS-only, portrait-only). These requirements are part of the definition of “shippable” and must be implemented across all V1 screens and interactions.

This chapter focuses on: (1) making the V1 feature set usable for people using assistive technologies (VoiceOver, Switch Control, larger text, reduced motion, etc.), and (2) ensuring the app is engineered so future translations/locales can be added without redesigning or rewriting core UI.

Non-goal: CH32 does not redefine product features, screens, plan gating, or interaction models. It adds accessibility + localization requirements on top of the owning chapters (e.g., Flow Builder behavior is owned by CH12).

2. Cross-references and dependencies

Accessibility requirements must be applied to every screen listed in **HZ-V1-CH05** and follow the design tokens/components in **HZ-V1-CH06**.

Key owning chapters referenced by this chapter:

- See: HZ-V1-CH02 (iOS-only, portrait-only constraint and general constraints).
- See: HZ-V1-CH04 (routes, modals, back behavior; impacts accessibility focus order).
- See: HZ-V1-CH06 (colors, typography, components, motion; impacts contrast and Dynamic Type).
- See: HZ-V1-CH08 (plan states; impacts paywall messaging and accessible gating).
- See: HZ-V1-CH12 (Flow Builder pan/zoom, branching, reorder; needs accessible alternative).

- See: HZ-V1-CH20/21/22 (Practice setup/active/history; timer controls and summaries).
- See: HZ-V1-CH25 (paywalls/upsells; accessible purchase flows).
- See: HZ-V1-CH31 (error states; accessible alerts and recoveries).

3. Accessibility goals and guiding principles

Handz must be usable without sight, without precise touch, and with larger text sizes. The app must not require color perception to understand state. The core value (building flows and drilling practice) must remain possible with assistive technologies.

Guiding principles (apply everywhere):

- No dead-ends
  - Every screen has a clear, reachable primary action and a reachable way back (or close).
  - If an interaction is gesture-only (e.g., pinch zoom), provide an alternate control or mode.
- No information conveyed by color alone
  - Selection or status must also be conveyed via icon/checkmark/border/label and via accessibility state/value.
  - Warnings must have text labels and accessible announcements (See: CH30/CH31).
- Predictable focus + reading order
  - Screen reader order follows visual hierarchy.
  - Modal open: focus moves to modal title. Modal close: focus returns to the triggering element.
- Bigger text must not break layout
  - Support iOS Dynamic Type sizes up to at least Accessibility Large. Content may scroll but must not clip/overlap.
  - Avoid fixed-height containers for text; prefer flexible height + scroll.
- Accessibility is part of V1 acceptance
  - Every feature acceptance test includes an a11y sub-test (VoiceOver + large text).

4. Global accessibility requirements (app-wide)

4.1 Touch targets, spacing, and gesture safety

- Minimum tap target: 44x44 pt for any interactive element (buttons, chips, icons, node handles).
- Provide at least 8 pt spacing between adjacent tap targets (or increase hit slop).
- Avoid requiring precision taps (users may wear gloves). Add invisible hit padding (hitSlop) for small icons.
- Gesture parity
  - If content is draggable/pannable/zoomable, provide alternate controls or an alternate mode.
  - Do not rely on long-press-only; if long-press exists, provide a visible menu button.
  - All destructive actions require a confirmation step with a clear cancel action (See: CH31).

4.2 VoiceOver semantics: labels, hints, values

Every interactive element must expose an accessibility label. Where needed, also expose a hint and value. Labels must be concise and specific.

- Label = what it is (e.g., “Add branch”).
- Hint = what it does (e.g., “Creates another response option from this move”).
- Value = current state when applicable (e.g., “Selected”, “Locked: Pro required”, “8 of 10 branches used”).
- Use correct accessibility roles (button, header, tab, switch, link).
- Decorative icons must not be focusable.

4.3 Dynamic Type + text sizing

- All text uses Dynamic Type styles (or equivalent scaling tokens in CH06).
- Body text must scale; headings may cap but must remain readable.
- Do not truncate critical labels. If unavoidable, provide full text in VoiceOver label and via expansion UI.

4.4 Color, contrast, and themes

Handz uses the CH06 palette. Ensure contrast meets accessibility standards for text and essential UI elements.

- Aim ≥4.5:1 contrast for normal text; ≥3:1 for large text and essential non-text indicators.
- Selection/focus cues must not be color-only (add checkmark/border/underline).
- Disabled state must be conveyed via opacity + accessibilityState.disabled.

4.5 Reduce Motion, haptics, and sound

- Respect iOS Reduce Motion: replace large animations with subtle fades.
- Avoid rapid flashing or excessive bounce.
- Haptics are optional and never the only cue.

4.6 Accessible notifications and alerts

- Blocking errors show accessible alerts with title, description, and recovery actions (See: CH31).
- Non-blocking messages use a toast/snackbar announced once (do not spam).
- Paywalls are announced as new screens; focus lands on paywall title.

5. Component-level accessibility specs (reusable building blocks)

Apply these semantics/behaviors consistently to all screens. Visual design is owned by CH06; this chapter adds accessibility behavior.

5.1 App shell: navigation bar, back/close, tab bar

- Navigation title uses accessibilityRole=header.
- Back button label includes destination when possible; otherwise “Back”.
- Close button label “Close” with hint “Dismisses this screen”.

- Tab bar exposes labels and selected state (icons alone not sufficient).

5.2 Buttons and icon buttons

- Icon-only buttons must have accessibilityLabel + hint and 44x44 hit area.
- Disabled buttons must remain readable and announced as disabled.
- Loading buttons set busy state and prevent double-taps.

5.3 Forms: fields, pickers, toggles

- Every field has a visible label (not placeholder-only).
- Validation errors are associated with fields and announced on submit.
- Pickers/steppers announce current value and allow step changes via VoiceOver.
- Required fields include “required” in label or hint.

5.4 Chips, tags, filters

- Chips announce selected/unselected state.
- Provide “Clear filters” action when filters active.
- If chips scroll horizontally, ensure VoiceOver can reach all items.

5.5 Lists, cards, and swipe actions

- Prefer explicit menu buttons over hidden swipe actions; if swipe exists, provide alternate.
- Cards group related info; label includes name + key metadata.
- Focus order: title -> primary action -> secondary actions.

5.6 Modals/bottom sheets/dialogs

- On open: focus goes to modal title. On close: return focus to trigger.
- Modals must have an accessible Close.
- Avoid nested modals; if unavoidable, each layer has its own close + focus containment.

5.7 Progress, timers, and live updates

- Timers show numeric time and do not rely on animation.
- Do not announce every second; announce key moments only (configurable later).
- Progress indicators expose accessible values (e.g., “Set 2 of 4”).

6. Screen-level accessibility checklist (applies to every CH05 screen)

Minimum bar for ‘Done’:

- Screen entry
  - Focus starts at screen title.
  - Back/Close is immediately discoverable.

    - Critical context announced once.
- Primary actions
    - Primary CTA is clearly labeled and reachable early in reading order.
- Empty states
    - Explain what’s missing and include an accessible CTA.
- Loading states
    - Accessible progress indicator with label; if >2s, show message and allow safe cancel/back.
- Error states
    - Follow CH31 patterns; blocking errors have recovery; focus lands on error title.
- Scrollable content
    - No clipping/overlap; header doesn’t cover content; long pages can scroll cleanly.

7. Flow Builder accessibility requirements (high-risk area)

Flow Builder (CH12) includes pan/zoom, branching, merges, and optional sequence details. CH32 requires full accessibility via an alternate representation and equivalent actions.

7.1 Two interaction modes: Canvas Mode and Outline Mode

- Canvas Mode (default)
    - Pan/zoom canvas with nodes and connectors.
    - Optimized for direct manipulation.
- Outline Mode (required)
    - Structured list representation of the same graph.
    - No pan/zoom or drag required.
    - Supports all core edits: replace root, add next, add branch (≤10), edit sequences, delete, reorder within a path, navigate merges/dangling endpoints.

Rule: Anything users can do in Canvas Mode must have an equivalent path in Outline Mode unless explicitly deferred to V2. If a feature cannot be made accessible in V1, it must be removed or deferred.

7.2 Mode switch UI

- Visible toggle labeled “Canvas” / “Outline” in Flow Builder header.
- Remember last mode per user (setting).
- Preserve selection when switching modes.

7.3 Node & sequence semantics (Canvas Mode)

- Move node label format
    - Label: “Move: {MoveName}”.
    - Value: include stance/side attributes if shown (e.g., “Orthodox, lead hand”).

- Hint: “Double tap for actions: replace, add next, add branch, details.”

- Node actions
  - Provide a visible ‘More actions’ button on each node.
  - Menu items: Replace move, Edit move details, Add next, Add branch, Edit sequence details (incoming/outgoing), Delete, Set as root (if allowed).
- Sequence connector (between moves)
  - Focusable element labeled: “Sequence: {FromMove} to {ToMove}”.
  - Value indicates whether sequence details exist and any condition label.

7.4 Canvas gesture alternatives

- Zoom
  - Buttons: Zoom In, Zoom Out, Reset/Fit, Center on selected, Center on root.
  - Announce zoom percentage changes without spam.
- Drag/reorder
  - Outline Mode provides Move Up/Down controls for reorder.
  - Canvas drag may exist but is not required for accessibility.
- Branch creation and limits
  - Branch add action shows current count and remaining capacity: “8 of 10 used”.
  - Branch list is accessible; each branch is a selectable row with label/condition.

7.5 Outline Mode information architecture (exact)

- Outline header
  - Flow name (header).
  - Actions: Replace Root, Add Path, Validate Flow, toggle Canvas/Outline.
- Path list
  - Each path shows: name, start move, end move, step count, merge/dangling status.
  - Selecting opens Path Detail.
- Path Detail
  - Ordered list of steps: Move row -> Sequence row -> Move row -> ...
  - Move row actions: Replace, Edit, Add next, Add branch, Delete.
  - Sequence row opens Sequence Detail Editor.
  - Reorder controls: Move Up/Down (where valid).
  - Branch/merge visibility: list outgoing branches; show incoming merges count at merge nodes.

7.6 Merges and dangling endpoints

- Dangling endpoint label: “Path ends here (no next move)” + action “Add next move”.

- Merge point label includes incoming count: “Merge: 3 incoming paths” + list of sources to navigate.
- Outline Mode supports ‘Go to merge source’ and ‘Go to merge destination’ actions.

7.7 Flow validation messages (recommended)

- Validation results list issues with severity (info/warn/error) and affected node/sequence.
- Each issue has a ‘Go to’ action; VoiceOver announces severity + summary.
- Example: “Warning: Move ‘Cross’ has 9 branches; adding more will hit the 10-branch limit.”

8. Practice Mode accessibility requirements

Practice behavior/gating is owned by CH20–CH22 and CH08. CH32 defines the a11y layer for practice setup, active session, and completion.

8.1 Practice Setup

- Selectable paths are reachable in list form (no canvas-only selection).
- Path selection announces selected state and order (first/second/etc.).
- Timer and assumed reps are adjustable with accessible pickers/steppers and announce values.
- If locked: show button “Unlock Practice (Pro)” and announce locked state.
- If credits exist: show “Use 1 credit” with remaining count announced.

8.2 Active Practice Session

- Glove-friendly controls
    - Large, spaced controls: Start/Pause/Resume, Next, Previous (if allowed), Skip Rest, End Session.
    - Confirm End Session.
- Announcements
    - Announce session start, set complete, rest start/end, session complete.
    - Do not announce every second. Optional last-10-seconds toggle later.
- Visible state
    - Show current path, step x/y, set x/y, remaining time.
    - VoiceOver label for status block includes these fields.
- Interruptions
    - If interrupted/backgrounded, show banner: “Session interrupted. Resume?” with Resume/End.

8.3 Completion + History

- Summary presents results as text first: duration, sets completed, paths drilled, assumed reps, mastery/maintenance progress.
- Breakdowns are focusable rows with clear labels.
- Actions: Continue, View History, Share Summary (if present) are clearly labeled.

9. Authentication, conversion prompts, paywalls (a11y)

Auth is owned by CH07; gating by CH08/CH25. CH32 ensures accessibility of these experiences.

9.1 Guest restrictions

- Guest cannot save flows; communicate before building starts.
- Account prompt must be accessible with explicit actions: Create Account, Log In, Cancel/Continue as Guest.
- Blocking modal used for gated actions (not a tiny toast). Focus lands on modal title.

9.2 Purchases

- Use StoreKit UI where possible; paywall content must be accessible.
- Paywall includes plan name, price, trial duration, renewal language, and actions: Start Trial/Subscribe, Restore, Not Now.
- On purchase success, return user to the feature they were trying to use (context preserved).

9.3 Upsell education pages

- Use headings so VoiceOver can navigate by sections.
- Avoid dense text; use bullets; provide text equivalents for charts.
- Links to references open in accessible in-app browser.

10. Offline and sync messaging (a11y)

Offline behavior is owned by CH28. CH32 defines accessible communication of offline status.

- Persistent banner when offline: “You’re offline. Some actions may not save.” Announce once.
- Offline failures show inline error + retry option.
- Uploads while offline show blocking message with recovery.

11. Localization readiness requirements

V1 may ship English-only, but the codebase must be localization-ready: no hardcoded strings; support plurals; locale-aware formatting; UI designed for text expansion.

11.1 Must-be-localizable content

- All UI strings, onboarding, tooltips, warnings, errors, paywall copy.
- All accessibility labels/hints/values.
- Pluralized strings and count-based labels.
- Dates/times and duration formatting.
- Notification titles/bodies (See: CH27).

11.2 Not localized by default

- User-generated content (names/notes).

- Internal canonical IDs/keys.
- Default move names ship in English for V1; users can rename/alias.

11.3 String key conventions

All strings referenced by stable keys; keys must not change once shipped.

Key scheme examples: **screen.{screen_id}.{element}.{purpose}** and **common.{component}.{element}**.

- Example keys: screen.flowBuilder.title, screen.practiceSetup.cta.unlockPro, common.error.networkOffline
- Do not concatenate strings; use placeholders/formatters.
- Use pluralization rules for: flows, sets, branches, credits, imports.

11.4 Translation expansion layout rules

- Assume translations are 30–50% longer; avoid fixed-width buttons; allow wrapping.
- Avoid text baked into images.
- Chip labels can scroll, but critical CTAs must remain visible.

11.5 Locale-aware formatting

- Use locale-aware date/time formatters; respect 12/24-hour settings.
- Numbers use locale separators automatically.
- Timer display uses mm:ss; summary uses localized duration strings.

12. RTL and directionality policy

Standard UI should mirror under RTL locales; Flow Builder canvas may remain LTR for V1 to preserve mental model. Outline Mode must mirror normally.

12.1 Standard UI mirroring

- Use leading/trailing constraints.
- Auto-mirror directional icons.
- Forms and lists mirror naturally.

12.2 Flow Builder directionality

- Canvas remains LTR under RTL locales in V1.
- Non-canvas UI mirrors normally.
- Outline Mode fully RTL-friendly.
- If full RTL canvas desired later, treat as explicit V2 project.

12.3 Placeholder registry (CH32-owned)

Placeholders (not locked yet) owned by CH32:

- PLACEHOLDER: Initial Localization Languages — Default: English-only V1; options include Spanish/Portuguese.

- PLACEHOLDER: RTL Canvas Support — Default: LTR-only V1 + evaluate for V2.
- PLACEHOLDER: Timer Announcement Policy — Default: key moments only; optional last-10-seconds later.

13. Accessibility QA workflow and test matrix

13.1 Required devices and settings

- Small iPhone size + large iPhone size (simulator acceptable).

- Settings to test
  - VoiceOver ON
  - Dynamic Type: Large and Accessibility Large
  - Bold Text ON
  - Increase Contrast ON
  - Reduce Motion ON
  - Alternate locales (English UK, Spanish) for formatting
  - RTL locale (Arabic) to validate mirroring policy

13.2 Required tools

- Xcode Accessibility Inspector (or equivalent).
- On-device VoiceOver testing.
- Color contrast checker.
- Optional snapshot tests for Dynamic Type.

13.3 When to run test passes

- During development: a11y check for each new screen before merging.
- Before beta: full pass on core paths (onboarding → build flow → save/share → practice → history).
- Before release: full pass and fix all P0/P1 issues.

13.4 A11y bug severity rubric

- P0 Blocker
  - Screen unusable with VoiceOver or large text; cannot proceed.
  - Critical action cannot be triggered without sight/precision touch.
  - Purchase/restore not accessible.
- P1 Major
  - Incorrect labels causing confusion but task still possible.
  - Focus order severely slows use.
  - Canvas-only feature missing Outline alternative.

- P2 Minor
  - Minor contrast or announcement issues that do not block completion.

14. Acceptance tests (a11y + localization readiness)

14.1 Flow Builder parity

- Given VoiceOver ON and Flow Builder opens
  - Then focus starts at title and the Canvas/Outline toggle is reachable immediately.
- Given Outline Mode is active
  - Then user can add branch, edit sequence, reorder steps, and navigate merges/dangling endpoints without gestures.
- Given Canvas Mode is active
  - Then nodes and connectors have labels, and zoom controls work without pinch.

14.2 Practice announcements

- Timer does not announce every second.
- Key moments announced once; controls are glove-friendly and at least 44x44.
- Interruptions show Resume/End banner on return.

14.3 Paywall accessibility

- Paywall focus lands on title.
- Start Trial, Restore, Not Now reachable and readable.
- After purchase, user returns to the originally attempted action.

14.4 Localization readiness

- No hardcoded user-visible strings; all via keys.
- Pluralization and locale-aware date/time formatting implemented.
- RTL locale does not break standard screens; Flow canvas remains LTR.

15. Replit Vibe-Coding Prompt — BUILD 32 (copy/paste)

Use this prompt inside Replit after the V1 project skeleton exists. It focuses only on accessibility + localization readiness (do not change product behavior).

BUILD 32 PROMPT (verbatim):

```
YOU ARE BUILDING: Handz iOS app (portrait-only). Implement CH32 Accessibility &
Localization Readiness requirements.

SCOPE:

1) Add accessibility semantics across all existing screens/components.

2) Add Outline Mode accessibility alternative for Flow Builder (required).
```

3) Make the app localization-ready (English-only content is OK, but no hardcoded strings).

4) Add a11y QA helpers and run a checklist.

DO NOT:

- Change product behavior defined in other chapters.

- Add new major features beyond a11y/localization needs.

DELIVERABLES:

A) Accessibility:

- 44x44 hit targets; add hitSlop where needed.

- VoiceOver labels/hints/values for buttons, chips, list items, nodes, sequences.

- Focus order fixes for navigation, modals, paywalls.

- Dynamic Type support up to Accessibility Large (no clipped text).

- Reduce Motion support for animations.

- Accessible alerts/toasts (announced once, not spammy).

B) Flow Builder:

- Add a toggle: Canvas / Outline.

- Outline Mode must let users: view and edit flow without pan/zoom/drag:

- Replace root

- Add next move

- Add branch (up to 10)

- Edit sequence details between steps

- Delete node/sequence

- Reorder steps (Move Up/Down) within a path where valid

- Discover merges (multiple incoming) and dangling endpoints

- Canvas Mode must include explicit Zoom In/Zoom Out/Reset/Fit and Center-on-selected/root buttons.

C) Localization Readiness:

- Create a localization system with stable keys.

- Replace ALL user-visible strings with localization keys.

- Implement pluralization helpers for counts (flows, sets, branches, credits, imports).

- Use locale-aware date/time formatting APIs.

- Ensure RTL locale doesn't break standard screens; keep flow canvas LTR per policy.

QA:

- Add a 'Developer Settings' debug screen with toggles to simulate:

- Dynamic Type size previews

- Reduce Motion

- Force RTL layout (if possible)

```
- Provide a short test script to run through core flows and an a11y punch list (P0/P1/P2).
```

16. Troubleshooting guide (common a11y + i18n issues)

16.1 VoiceOver issues

- VoiceOver reads generic labels
  - Add accessibilityLabel/hint for icon-only elements.
- Focus jumps in modals
  - Set initial focus to title; trap focus inside; return focus on close.
- Canvas not usable
  - Ensure Outline Mode exists and supports core edits.
- Timer spams announcements
  - Throttle announcements; announce key moments only.

16.2 Dynamic Type layout breakage

- Remove fixed heights; allow containers to grow; enable scrolling.
- Allow CTA labels to wrap or use shorter localized variants.
- Use stacked layouts for very large text sizes.

16.3 Localization issues

- Remove manual string concatenation; use formatted localized strings.
- Use proper plural rules instead of adding “s”.
- Use locale-aware date/time/number formatters.

16.4 RTL issues

- Replace left/right with leading/trailing.
- Ensure directional icons auto-mirror.
- Lock Flow canvas direction to LTR; keep Outline mirrored.

17. Definition of Done (CH32)

- All screens pass VoiceOver navigation (no P0 blockers).
- All screens render at Dynamic Type Accessibility Large without clipping critical text.
- Flow Builder Outline Mode exists with core edit parity.
- All strings are via localization keys; plurals + locale formatting implemented.
- RTL locale does not break standard UI; Flow canvas remains LTR; Outline mirrors.
- Acceptance tests in §14 executed and documented.