

HZ-V1-CH19 Import Conflict Resolution

Doc ID:	HZ-V1-CH19_Import_Conflict_Resolution_R1
Revision:	R1 (2026-01-02)
Status:	Draft
Depends on:	CH08 (Entitlements & Plan States), CH10 (Canonical IDs/Aliases/Families/Variants), CH18 (Inbox), CH17 (Share Links), CH29 (Media & Limits), CH11 (Revert Model)
Related:	CH05 (Screen Inventory), CH16 (Flow Detail), CH31 (Error States), CH33 (Analytics), CH30 (Safety/Abuse Limits)
Supersedes:	—
Owned Decisions:	Import preflight and conflict-resolution UX, mapping/merge rules for moves and move metadata during import, persistence/audit for imports.
Open Questions / Placeholders:	PLACEHOLDER: Import batch size cap • Owner: CH30; PLACEHOLDER: Pro-only overwrite rules granularity • Owner: CH08; PLACEHOLDER: Canonical matching heuristics thresholds • Owner: CH10.

This chapter specifies exactly how Handz handles **imports** (accepting a shared flow into the recipient’s Inbox) when the incoming flow references moves or move details that do not cleanly match the recipient’s current move library. The goal is **zero guesswork**: deterministic mapping rules, minimal-but-safe user choices, and a predictable audit/revert model.

1. Scope and Goals

CH19 covers only the **import conflict resolution layer**. Link creation, inbox caps, practice gating, and share lifecycle are owned elsewhere.

- **In scope:** preflight analysis, move mapping, missing-move handling, sender customizations vs recipient library, safe defaults, import persistence, and UI flows for resolving conflicts.
- **Out of scope (cross-referenced):** share link creation/revocation (CH17), inbox list UI and caps (CH18), entitlements (CH08), media storage/2GB cap and link-vs-upload rules (CH29), warnings/abuse thresholds (CH30).
- **Primary success criteria:** (1) recipient can import and view a flow without breaking rendering; (2) resolution choices are understandable in under 30 seconds; (3) no silent overwrites; (4) revert is always possible.
- **V1 product intent:** keep this simple. No “add both” branching variants, no custom messages, and no multi-step wizard unless necessary.

2. Definitions

- **Import Package:** the structured payload created when someone shares a flow. It includes the flow graph + required move descriptors so the recipient can render it even if they don't have every move.
- **Recipient Library:** the recipient's saved moves and their canonical records.
- **Flow-local Move:** a move definition stored inside an imported flow only (does not alter the recipient's library). Used for missing moves or sender-specific customizations when recipient chooses not to overwrite.
- **Canonical Move ID:** stable identifier used to map the "same" move across users (owned by CH10).
- **Alias:** alternate name string that may map to the same canonical move family (owned by CH10).
- **Conflict:** any situation where Handz cannot map an incoming move cleanly to a recipient library move without either (a) losing information or (b) overwriting user data.
- **Preflight:** the review step shown before saving an import, summarizing conflicts and required choices.

3. Conflict Types (Authoritative)

Handz recognizes these conflict classes. The UI and resolution rules are based on class.

- **C1 Missing Move:** incoming move cannot be matched to any recipient library move via canonical ID or alias match.
- **C2 Ambiguous Match:** incoming move matches multiple recipient moves (e.g., two user-created "Jab" variants) or multiple canonical candidates.
- **C3 Sender Customization:** canonical match exists, but incoming move includes sender-specific metadata that differs (notes/tags/attributes/media links).
- **C4 Media Incompatibility:** incoming move references *uploaded media* that is not shareable (V1 rule: uploaded files are private-only; only links are shareable) — import must degrade gracefully.
- **C5 Schema Version Drift:** import package schema version is older/newer than app version; fields missing or extra must not break import.

4. Import Lifecycle (From Link → Inbox → Library)

- **Step 0: Open link** (CH17). App resolves link → fetches import package.
- **Step 1: Preflight** (this chapter). App parses package, runs mapping, classifies conflicts.
- **Step 2: Save to Inbox** (CH18). Import is stored as an Inbox item with a stable import_id and a "resolution snapshot".
- **Step 3: View imported flow.** Rendering must work even if unresolved (using flow-local moves).
- **Step 4: Save to Library** (CH18). Requires account and is subject to plan caps (Free: 2 saved flows). If cap exceeded, the user must manage or upgrade before saving.

Important V1 lock: Free users can accept imports into Inbox (cap 10) and view them, but cannot practice Inbox items. Practice credits apply only to saved flows.

5. Mapping Algorithm (Deterministic)

Goal: map each incoming move reference to either (A) an existing recipient move ID or (B) a flow-local move definition.

5.1 Inputs

- Incoming move descriptor: {canonical_id?, primary_name, aliases[], family_id?, variant_of?, attributes?, user_notes?, tags?, media_links[], uploaded_media_refs?}
- Recipient move index: all recipient moves with their canonical_id (if any), primary_name, aliases, and a normalized token set for fuzzy comparison.

5.2 Matching precedence (stop at first success)

- **P1 Canonical ID exact match:** if incoming.canonical_id matches a recipient move.canonical_id → map to that move.
- **P2 Explicit alias match:** if any incoming.alias equals a recipient.primary_name or recipient.alias → candidate match.
- **P3 Normalized name match:** compare normalized tokens (lowercase, trim, remove punctuation) of incoming.primary_name to recipient.primary_name and aliases.
- **P4 Family-based fallback:** if incoming.family_id exists and recipient has moves in that family, pick the best normalized-name match within that family; otherwise mark ambiguous.
- **P5 No match:** classify as C1 Missing Move.

5.3 Ambiguity rules

- If P2–P4 yield exactly 1 candidate → map to it.
- If multiple candidates tie → classify as C2 Ambiguous Match.
- If recipient has multiple local variants with same name (user-created), do not auto-pick; require user choice in preflight.

5.4 Output mapping object

For each incoming move_ref_id (the node reference used in the flow graph), produce:

```
{ move_ref_id, resolution: 'mapped'|'flow_local'|'needs_choice', recipient_move_id?,  
  flow_local_move_id?, conflict_class?, notes? }
```

6. Preflight UX (What the User Sees)

Preflight appears when the user taps **Accept Import** from a share link, before saving to Inbox. It should be fast, scannable, and safe by default.

6.1 Preflight screen layout

- **Header:** "Review Import" + flow name + sender handle (if available).
- **Summary counters:** Moves: X total • Conflicts: Y • Media issues: Z.
- **Primary CTA:** [Save to Inbox] (disabled if required choices not completed).
- **Secondary:** [Cancel].
- **Optional:** [View Flow Preview] opens read-only render using flow-local placeholders.

6.2 Conflict cards

Each conflict produces one card. Cards are grouped by class: Missing, Ambiguous, Sender Customizations, Media.

- **Card header:** incoming move name + badge (Missing / Choose / Customize / Media).
- **Card body:** short explanation + the *two* allowed actions (keep simple).
- **Default selection:** safest choice preselected (never overwrites library).
- **Required vs optional:** Missing + Ambiguous are required to resolve; Sender Customizations are optional (defaults to ignore).

6.3 Allowed user actions (V1 simplified)

User requested: no custom message prompt and no "add both" option. Therefore actions are:

- **Action A — Use My Library:** map to an existing recipient move (keeps recipient's move as-is).
- **Action B — Use Sender Version (Flow-only):** create/keep a flow-local move definition inside this imported flow. Does not alter recipient library.
- **Action C — Overwrite My Library:** *optional and gated*. Only available if plan state allows (likely Pro; owned by CH08). If enabled, it replaces recipient move metadata with sender metadata. Must include a clear warning and a one-tap revert entry.

6.4 Bulk controls

- **Apply to all (Sender Customizations only):** "For customized moves, default to: [Use My Library]".
- No bulk apply for ambiguous matches (too risky).
- If user chooses "Use Sender Version" for multiple missing moves, they are created as flow-local moves in one batch.

7. Resolution Rules by Conflict Class

7.1 C1 Missing Move

- Default: **Use Sender Version (Flow-only)** (creates a flow-local move).
- If the incoming move provides canonical_id and the recipient has that canonical family but no exact match, show Action A with a picker of close candidates.
- If the user selects Action A, the node maps to the chosen recipient move; incoming move descriptor is kept only as a "source note" in the import audit record (not shown by default).

7.2 C2 Ambiguous Match

- Default: no selection; user must choose.
- Show a picker list of candidate recipient moves with: name, family, user tags, and a tiny preview of any attached link thumbnail (if exists).
- Provide Action B as escape hatch: "Use Sender Version (Flow-only)" to avoid choosing.

7.3 C3 Sender Customization

- Default: **Use My Library** (ignore sender differences).
- If user selects Action B (Flow-only), show a concise diff preview (fields that differ) but do not force editing.
- If Action C (Overwrite) is available, require a confirmation sheet: "This will update your move: . You can revert later."

7.4 C4 Media Incompatibility

- V1 rule: uploaded media is private-only and **cannot be shared**; only links are shareable (CH29).
- If incoming references uploaded media, import must drop it and show a badge: "Video not included (upload is private)."
- If incoming includes a shareable link, keep it.
- Never block import solely due to media issues.

7.5 C5 Schema Version Drift

- Unknown fields are ignored but preserved in raw payload for future compatibility.
- Missing optional fields default to empty.
- If required fields missing (flow graph invalid), block save-to-inbox and show error state with a support code.

8. Plan State Gating (Guest/Free/Pro)

- **Guest:** cannot save flows; cannot save imports to Inbox. If a guest opens a share link, they may preview a demo render, but must create an account to accept.
- **Free:** Inbox cap 10. Free can accept imports into Inbox and view them. Free cannot practice Inbox items. Free can save imports into Library only if within saved-flow cap (2).
- **Pro/Trial:** normal import; overwrite action (Action C) may be enabled; share link caps differ.

If Free user is at **2 saved flows** and attempts "Save to Library" from an import: show the cap screen with options: [Manage Flows] (delete/upgrade) and [Upgrade]. Do not silently replace.

9. Data Model and Persistence

9.1 Import package schema (V1)

```
{ "schema_version": "1.0", "share_id": "<string>", "created_at": "<iso8601>", "sender": {
  "user_id": "<uuid?>", "handle": "<string?>", "display_name": "<string?>" }, "flow": {
  "flow_id": "<uuid or sender-local id>", "name": "<string>", "description": "<string?>",
  "nodes": [ { "node_id": "<string>", "type": "move", "move_ref_id": "<string>", "position":
    { "x": 0, "y": 0 }, "meta": { ... } }, { "node_id": "<string>", "type": "sequence_optional",
    "sequence_ref_id": "<string>", "meta": { ... } } ], "edges": [ { "edge_id": "<string>",
    "source_node_id": "...", "target_node_id": "...", "kind": "linear|branch", "label": "<string?>",
    "sequence_ref_id": "<string?>", "style": "solid|dashed" } ] }, "move_descriptors": [ {
  "move_ref_id": "<string>", "canonical_id": "<string?>", "primary_name": "<string>", "aliases":
    [ "<string>" ], "family_id": "<string?>", "variant_of": "<canonical_id?>", "attributes": {
    "stance": "<...?>", "side": "<...?>", "targets": [ "head" ], "tags": [ "..." ] }, "user_notes":
    "<string?>", "media_links": [ "<url>" ], "uploaded_media_refs": [ "<opaque>" ] } ] }
```

9.2 Inbox item persistence

- Inbox items store: import_id, received_at, sender, flow_snapshot (graph), resolution_snapshot, conflict_summary counts, and a raw_payload blob for future compatibility.
- Resolution snapshot stores chosen actions per move_ref_id plus any user selections for ambiguous matches.
- When user saves to Library, we persist a new flow record with embedded flow-local moves (if any) and a pointer back to import_id for audit/revert.

9.3 Audit + revert hooks

- Every import creates an audit record: who sent it, when, and what resolution actions were applied.
- If overwrite (Action C) is used, create a per-move backup snapshot enabling revert of that move only (not whole library). This references CH11 revert model.
- Expose "Revert changes from import" entry point on the import detail screen.

10. Error States and Required Copy

Exact strings can move to CH15 Copy Pack later, but CH19 defines the required messages and where they appear.

10.1 Blocking errors

- **Invalid package:** "This link can't be imported right now." Subtext: "The shared flow is missing required data." CTA: [Close] + [Report Problem].
- **Network failure:** "Connection lost" Subtext: "Check your internet and try again." CTA: [Try Again] [Cancel].
- **Guest trying to accept:** "Create an account to save" Subtext: "Guests can preview, but importing requires an account." CTA: [Sign up] [Log in] [Cancel].

10.2 Non-blocking warnings

- **Media dropped:** "Video not included" Subtext: "Uploads are private. Ask the sender for a link."
- **Customizations ignored:** "Using your existing move details" Subtext: "Sender notes won't overwrite your library."

10.3 Cap/gating messages

- **Inbox full (Free cap 10):** "Inbox is full" Subtext: "Free accounts can hold up to 10 imports." CTAs: [Manage Inbox] [Upgrade].
- **Saved flows full (Free cap 2):** "You've reached 2 saved flows" Subtext: "Delete one or upgrade to save more." CTAs: [Manage Flows] [Upgrade].

11. Analytics Hooks (High-Level)

- import_opened (from link)
- import_preflight_shown
- import_conflict_count {missing, ambiguous, customized, media}
- import_action_selected {actionA|actionB|actionC} per conflict class
- import_saved_to_inbox
- import_save_to_library_attempted
- import_save_to_library_blocked_cap
- import_overwrite_confirmed
- import_revert_used

Event names and properties are finalized in CH33.

12. Acceptance Tests (Given / When / Then)

- **Missing move default:** Given an import where a move has no match, when preflight loads, then that move shows as Missing with Action B preselected and the flow preview renders without crashing.
- **Ambiguous requires choice:** Given an import where a move matches two recipient moves, when preflight loads, then Save to Inbox is disabled until the user selects Action A with a specific candidate or selects Action B.
- **No silent overwrite:** Given a customized move with sender notes, when user keeps default Action A, then recipient library move is unchanged after import and audit record notes "customizations ignored".
- **Media drop:** Given an import with uploaded_media_refs only, when saved to inbox, then the flow shows "Video not included" badge and no broken thumbnail is displayed.
- **Guest gating:** Given a guest user opens a share link, when they tap Accept, then they are routed to Sign Up/Log In and the import is not saved.
- **Free inbox cap:** Given a Free user has 10 inbox items, when they attempt to accept another import, then they see Inbox Full screen with Manage/Upgrade CTAs and import is not saved.
- **Free saved flows cap:** Given a Free user has 2 saved flows, when they try Save to Library from an imported flow, then they see the cap screen and cannot proceed without deleting or upgrading.
- **Overwrite confirmation:** Given overwrite is available, when user selects Action C, then an explicit confirmation is shown and a per-move backup snapshot is created.
- **Revert per move:** Given overwrite occurred, when user taps Revert on that move, then only that move's metadata is restored and the imported flow remains intact.

13. Replit Build Prompt (Implement CH19 Only)

You are implementing Handz V1 PRD Bundle chapter CH19 only (Import Conflict Resolution). Inputs: share link -> import package payload -> preflight screen -> save to inbox. Do NOT implement CH17/CH18/CH10 fully; stub their APIs and respect cross-references. Build: 1) Data parsing for ImportPackage schema_version 1.0; store raw_payload. 2) Matching engine implementing precedence P1-P5 and ambiguity rules (CH19 §5). 3) Conflict classification C1-C5; produce mapping objects per move_ref_id. 4) Preflight UI: - Summary counters - Conflict cards grouped by class - Allowed actions A/B (+ C if entitlements allow; stub entitlements check) - Disable Save until required choices resolved 5) Save-to-inbox action persists: import_id, flow_snapshot, resolution_snapshot, conflict_summary. 6) Ensure flow preview renders using: - recipient move if mapped - flow-local move if chosen/required - media drop behavior for uploaded_media_refs 7) Provide hooks for later features: - overwrite confirmation sheet - per-move backup snapshot for revert (stub storage) 8) Add unit tests for mapping and conflict classification and UI tests for gating. Stop and surface a 'PRD Assumptions' block for any missing dependency details.

14. Troubleshooting Notes (CH19)

- **Symptom:** Imported flow renders with blank nodes → **Check:** mapping output contains neither recipient_move_id nor flow_local_move_id. Ensure every move_ref_id resolves to one.
- **Symptom:** Save button never enables → **Check:** ambiguous conflicts marked required are missing selection; verify UI state updates after user picks a candidate.
- **Symptom:** Duplicate moves appear after import → **Check:** you accidentally wrote flow-local moves into global library. Flow-local moves must be scoped to the imported flow record only.

- **Symptom:** Media thumbnails break → **Check:** uploaded_media_refs are being treated as URLs; they must be dropped and replaced with a badge.
- **Symptom:** User claims their move changed unexpectedly → **Check:** overwrite path triggered without confirmation; Action C must always confirm and log audit + backup snapshot.