

ISO 20022 pacs.008 Structured-Address Migration Checklist

A 23-point project tracker for the November 2026 CBPR+ deadline

Sebastien Rousseau

May 2026

Contents

ISO 20022 pacs.008 Structured-Address Migration Checklist	1
Why this matters	1
Section A — Inventory (do these first)	1
Section B — Mapping rules	2
Section C — Generator updates	2
Section D — Receiver fallback	3
Section E — End-to-end test harness	3
Section F — Cutover signalling	3
Section G — Post-cutover	4
How to use this checklist	4

ISO 20022 pacs.008 Structured-Address Migration Checklist

A 23-point project tracker for the November 2026 CBPR+ deadline.

Why this matters

On **22 November 2026**, SWIFT’s CBPR+ usage guidelines move pacs.008 into a **structured-address-mandatory** regime for the cross-border payments corridor. Messages that still rely on **AdrLine** free-text fallback will be **rejected at the network layer** — not warned about, not best-effort downgraded. Counterparty banks that haven’t updated their pacs.008 generator by cutover stop being able to instruct correspondent payments.

The migration is not technically difficult. It is operationally fiddly. The data lives in legacy customer-master systems that have modelled “address” as one free-text blob for thirty years. Pulling TwnNm, PostCode, Ctry, BldgNb, StrtNm out of that blob — reliably, with a fallback path for ambiguous records — is the work.

This checklist is what I’d hand a migration squad on day one.

Section A — Inventory (do these first)

1. Pull last 90 days of outbound pacs.008. Sample by corridor, by amount band, by counterparty class. You want representative volume across all your customer segments before you draw any conclusions

about what residue remains.

2. Run a Pst1Adr audit. For every sampled message, classify the Pst1Adr element into one of four buckets: (a) fully structured, (b) partially structured + AdrLine residue, (c) AdrLine only, (d) missing entirely. The size of bucket (c) is the size of your migration backlog.

3. Trace every (c)/(d) message to a source system. The fix never goes in pacs.008 itself. It goes in whichever upstream system populated the originating instruction (corporate-banking portal, host-to-host gateway, file-driven batch processor, mainframe payments hub). Identify them all.

4. Map customer-master fields to ISO 20022 slots. Document, source system by source system, which field in the customer-master maps to TwnNm, PostCode, Ctry, BldgNb, StrtNm, Dept, SubDept, CtrySubDvsn. Where a mapping doesn't exist, flag that record class — it's the long-tail you'll spend Q3 on.

Section B — Mapping rules

5. Decide your AdrLine fallback policy. CBPR+ permits a *hybrid* model — structured fields + a single AdrLine for residue — until November 2026. Decide whether you emit structured-only from day one (cleaner, slightly higher parse-failure rate) or hybrid until cutover (safer, more post-migration cleanup later).

6. Define your address-parser ruleset. For records where the customer-master is unstructured free text, write deterministic parse rules: regex-driven for known formats (UK postcodes, US ZIP, Singapore postal, etc.), country-aware fallback for everything else. Document every rule with a test input/output pair.

7. Country-code normalisation. Every Pst1Adr MUST carry a Ctry two-letter ISO 3166-1 code. Audit your customer-master for country values held as full names, three-letter codes, or local-language equivalents. Build a canonicalisation table once.

8. Truncation policy. TwnNm is capped at 35 characters. StrtNm is capped at 70. Decide what happens when a real-world value exceeds the cap — truncate, shift overflow to AdrLine, or reject upstream. Write the policy down before engineering picks one.

Section C — Generator updates

9. Patch the pacs.008 templating layer. Whatever produces your outbound pacs.008 XML (Volante, IBM Sterling, in-house Java/.NET, mainframe Cobol) — find the template that emits Pst1Adr and add the structured-field branches. Every existing message path must call through it.

10. Unit test every field. The eight sub-fields (TwnNm, BldgNb, StrtNm, PostCode, Ctry, Dept, SubDept, CtrySubDvsn) each need a positive test (present, correctly populated) and a negative test (absent, fallback behaviour correct). Test the truncation policy from item 8.

11. Validate with the CBPR+ usage guidelines XSD. Don't trust your generator's internal validation. Run every unit-test output through the published CBPR+ XSD before you call it green. The XSD is the only contract that matters at cutover.

12. Schedule generator deployment to all sending channels. Corporate portal, host-to-host, file-driven, treasury workstation, correspondent gateway — every channel that emits pacs.008 needs the

patched generator. Sequence the rollouts to keep the legacy path alive on at least one channel until you have field confirmation the new one is working.

Section D — Receiver fallback

13. Keep your inbound parser hybrid-tolerant until 2027. Even after November 2026, you will receive pacs.008 from counterparties that haven't migrated — non-CBPR+ corridors, domestic schemes, slow movers. Your receiver pipeline must continue to parse `AdrLine`-only messages cleanly through 2027 at least. Don't strip that code yet.

14. Add a counterparty-readiness register. Track, by BIC, whether each counterparty has confirmed structured-only readiness, hybrid, or unmigrated. Use it to route exception handling and to escalate to the scheme manager where needed.

15. Build a structured-vs-unstructured KPI dashboard. Per-counterparty, per-week, percentage of inbound pacs.008 that carries a fully structured `Pst1Adr`. The graph going up to 100% is your migration progress. The plateau below 100% is your remaining counterparty risk register.

Section E — End-to-end test harness

16. Build a corridor-by-corridor test matrix. SWIFT cross-border, SEPA SCT Inst, CHAPS RTGS, T2/T2S, FedNow, domestic ACH ISO 20022 corridors — each has slightly different usage guidelines. One test case per (corridor × address-class) cell of the matrix. The matrix is your acceptance gate.

17. Test the truncation edge cases. Real-world addresses break the 35/70-char caps regularly. Run targeted test cases for: a 40-char `TwnNm`, an 80-char `StrtNm`, a multi-line address with a `Dept` element, a record where `PostCode` is missing entirely (some countries don't use them).

18. Test the canonicalisation table. Feed your address-parser the country values “United States”, “USA”, “US”, “U.S.A.”, and confirm all four produce `Ctry=US`. Repeat for every country class you've identified in your customer-master.

Section F — Cutover signalling

19. Coordinate with your scheme manager. SWIFT publishes activation windows per CBPR+ phase. Confirm your activation slot at least eight weeks before cutover and get the scheme manager's sign-off in writing. The activation window is the only date that matters; everything else sequences off it.

20. Brief your operations team on the failure mode. Post-cutover, an unmigrated pacs.008 fails at the SWIFT layer with a specific reject code. Make sure your operations bridge knows what the reject code looks like, where it surfaces in your operations console, and which engineering on-call owns the fix path. A scheme-level reject is not the time to be discovering this.

21. Plan a rollback path. Have a documented, tested procedure for reverting the generator to hybrid mode if something unexpected happens in the first 48 hours post-cutover. Test it in dilution before cutover, not after.

Section G — Post-cutover

22. KPI reporting, weekly, for at least six weeks. Bytes-on-the-wire ratio of structured to unstructured per counterparty, per week. Reject-rate trend. Mean-time-to-resolve on address-parsing exceptions. Six weeks of clean numbers is the bar for declaring the migration done.

23. Update your customer-master capture rules. The pacs.008 deadline doesn't change customer-master schemas on its own. Make sure new customer onboarding captures structured fields at source, so the migration doesn't have to be repeated for every new customer record created post-cutover.

How to use this checklist

Print it. Tape it to the wall in your migration squad's room. Cross items off as they complete. Run a daily standup against the open items.

The migration is a project, not a deploy. Treat it like one.

This checklist is distilled from EPAA working-group output, SWIFT CBPR+ usage guidelines, and the RedCompass 308-bank ISO 20022 readiness survey. Errors are mine.

If your bank is behind, [get in touch](#) — I do selective ISO 20022 transformation consulting and can typically shorten a residual programme by 4-6 weeks with the right intervention point.

Sebastien Rousseau — May 2026 <https://sebastienrousseau.com>