

Parquet-to-Iceberg Migration Control Pack

A preflight, validation, cutover, and rollback workbook

Prepared by **Vertex Frontier**

08 16, 2026

Use this pack before running `migrate`, `snapshot`, or `add_files`.

The workbook is designed to make table identity, file ownership, validation evidence, and rollback decisions visible to the whole platform team.

Contents

1	How to use this control pack	1
2	1. Source estate classification	1
3	2. Preflight gate	1
4	3. Procedure decision matrix	2
5	4. Migration control plane	3
6	5. Validation scorecard	3
7	6. Cutover planner	4
8	7. Risk register	5
9	8. Post-migration operating contract	5
10	Final acceptance	6

1 How to use this control pack

A Parquet-to-Iceberg migration is not complete when a procedure returns successfully. It is complete when the team can explain what the target table owns, which writers can commit, which cleanup jobs can delete, what evidence proves equivalence, and how the known-good state can be restored.

Complete the pack in order:

1. Classify the source estate.
2. Pass the schema, partition, writer, and ownership preflight gates.
3. Select a procedure and record why the alternatives were rejected.
4. Run validation at file, partition, aggregate, query, and engine levels.
5. Choose a cutover pattern and write down the rollback trigger.
6. Assign post-migration maintenance ownership before declaring success.

Operating rule: A metadata commit is one event in a migration, not the acceptance criterion.

2 1. Source estate classification

The word “Parquet” does not tell you whether a migration is straightforward. Use the matrix below to record the state that best describes the table.

Code	Observed state	Evidence	Initial implication
A	Hive-registered Parquet	Catalog definition, schema, partitions, and location are discoverable.	Snapshot or migrate may fit after writer and catalog checks.
B	Raw Hive-style Parquet	Paths encode values such as event_date=2026-08-16.	Define and validate a source table before adoption.
C	Raw non-Hive-partitioned Parquet	Partition values are not reliably encoded in paths.	Use explicit mapping or a rewrite; hold add_files until proven safe.
D	Mixed or dirty dataset	Multiple schemas, writers, layouts, or ownership boundaries.	Normalize the contract or choose a controlled rewrite.

Record your classification:

Evidence location:

3 2. Preflight gate

Mark each gate only after the evidence exists. “The engine will probably handle it” is not evidence.

Gate	Acceptance question	Status / evidence
Schema inventory	Have all distinct writer and partition families been profiled?	

Gate	Acceptance question	Status / evidence
Field identity	Are field IDs, name fallback, renames, and duplicate names understood?	
Type compatibility	Are numeric, timestamp, decimal, and nullability changes reviewed?	
Partition mapping	Can each partition value be recovered from paths, files, or an explicit definition?	
Writer inventory	Are batch, streaming, crawler, notebook, and repair jobs listed?	
Reader inventory	Are catalog readers and direct path readers listed?	
Ownership	Who can write, delete, compact, expire snapshots, and drop tables?	
Rollback	What known-good source, backup, or snapshot will be retained?	

Preflight decision:

4 3. Procedure decision matrix

Use the procedure that matches the ownership and isolation you actually need.

Path	Use it when	Avoid it when	Primary risk
Snapshot	You need a new Iceberg table while the source remains available for testing or staged transition.	The source will keep changing without a synchronization or cutover plan.	Target drifts from source changes.
Migrate	You can stop writers and want to replace the source table identity.	Legacy writers cannot stop or the catalog cannot support replacement safely.	Readers or writers break at the identity switch.
Add files	The target schema is defined and known files must be adopted or added after an initial migration.	Schema, partition mapping, duplicate behavior, or ownership is uncertain.	Wrong partitions or duplicate rows.
Full rewrite / shadow	You need isolation, deduplication, new partitioning, or better file sizing.	There is no compute, storage, or staged reconciliation window.	Higher cost and longer reconciliation.

Selected path:

Reason and rejected alternatives:

Catalog and engine versions tested:

5 4. Migration control plane

For a single experiment, a notebook may be enough. For a production estate, record one stateful record per table. The control plane may be a relational table, workflow system, key-value store, or service, but it needs the same fields.

Field	Purpose	Value / evidence
Idempotency key	Prevents a retry from creating a second target or re-registering the same files.	
Phase	Inventory, preflight, import, validation, cutover, rollback, or complete.	
Source and target	Makes physical and catalog ownership unambiguous.	
Attempts and last error	Separates transient failures from deterministic data or permission failures.	
Evidence URI	Points to inventory, validation results, query diffs, and approvals.	
Approver	Records who accepted the evidence and decision.	

Suggested idempotency key:

Current phase:

6 5. Validation scorecard

A global row count can match while one partition is missing and another contains duplicates. Validate from broad coverage to targeted business evidence.

Check	Compare	Hold condition	Status
File coverage	Expected source files versus Iceberg-referenced files.	Unexplained missing, duplicate, or out-of-scope file.	
Partition counts	Counts by every important partition.	Mismatch without approved explanation.	
Aggregates	Sums, minima, maxima, and distinct counts for critical measures.	Unexpected drift, overflow, or coercion.	
Null rates	Keys, partitions, and frequently filtered columns.	Unexpected increase or newly possible null.	

Check	Compare	Hold condition	Status
Duplicate keys	Business keys or event IDs grouped and counted.	Duplicates introduced by overlap or re-registration.	
Query diffs	Representative production queries and result checksums.	Different result set or failed predicate.	
Engine reads	Every production query engine and supported reader.	Field-resolution or catalog error.	
Rollback behavior	Known-good snapshot or backup can be restored.	Rehearsal fails or evidence is missing.	

Validation query notes:

Evidence folder:

7 6. Cutover planner

Choose a cutover pattern that matches writer control, downtime tolerance, and rollback strength.

Pattern	Sequence	Rollback strength	Prerequisite
Freeze-and-switch	Stop writers; migrate; validate; redirect readers and writers; observe the backup period.	High	Complete writer inventory and maintenance window.
Dual-write	Backfill; write to old and new; reconcile; switch after an observation window.	Medium	Stable event IDs and reconciliation process.
Blue/green	Blue remains current; green is validated; move traffic in stages.	High	Separate catalog, permissions, and monitoring tests.
Shadow rewrite	Rewrite physically; run shadow queries; switch after acceptance.	High	Storage, compute, and reconciliation budget.

Selected cutover pattern:

Writer redirect time:

Reader redirect time:

Observation window ends:

Rollback owner:

8 7. Risk register

A useful risk register connects a red flag to evidence and a decision. It should be reviewable by someone who did not run the migration.

ID	Risk	Evidence to collect	Decision
R-001	Unknown source writer	Scheduler list, IAM review, crawler and repair-job inventory.	Go / Hold / Rollback:
R-002	Unrecoverable partitions	Path inventory and explicit source-table definition.	Go / Hold / Rollback:
R-003	Schema identity uncertainty	Schema report and engine compatibility tests.	Go / Hold / Rollback:
R-004	Duplicate registration	File list, duplicate check, and row-key test.	Go / Hold / Rollback:
R-005	Unsafe cleanup authority	IAM, lifecycle, retention, and deletion review.	Go / Hold / Rollback:
R-006	Validation drift	Partition diffs, checksums, and query outputs.	Go / Hold / Rollback:
R-007	Rollback not rehearsed	Rehearsal record and snapshot or backup ID.	Go / Hold / Rollback:

9 8. Post-migration operating contract

Migration changes who can manage the table. Record the owner and guardrail for each operation before the source backup is deleted.

Operation	Purpose	Guardrail
Expire snapshots	Remove snapshots and files no longer needed for time travel or rollback.	Retention covers recovery, late jobs, and the agreed rollback window.
Remove orphan files	Clean files not referenced by table metadata.	Dry-run first; retention exceeds maximum write duration.
Rewrite data files	Compact small files and reduce file-open overhead.	Do not confuse compaction with partition redesign or data-quality repair.
Rewrite manifests	Regroup metadata entries to improve planning.	Measure planning behavior before and after.
Rollback rehearsal	Prove the team can return to a known-good state.	Rehearse before deleting the source backup or shortening retention.

What migration does not fix: In-place registration can preserve small files, poor distribution, skewed partitions, duplicate business keys, stale schemas, and an unsuitable partition strategy. Use migration to change table semantics and ownership; use rewrites and maintenance operations to change physical quality.

10 Final acceptance

Fill this section only when the evidence is attached.

Target table identifier:

Current snapshot or commit:

Validation evidence location:

Rollback reference:

Decision:

Approver:

Date:
