

# Leaving Power Without Rewriting

## *A CIO's Guide to IBM i Modernization*

Dr. Bruce Acacio

Infinite Corporation · June 2026

### **Abstract**

For four decades the IBM i (AS/400) platform has run the operational core of manufacturers, distributors, insurers, and public agencies. The applications remain correct and valuable; the platform beneath them, however, increasingly constrains the enterprise through proprietary hardware cycles, rising licensing, and a contracting skills market. This guide is written for the CIO who must decide what to do about IBM i without betting the business on a multi-year rewrite. It frames the four strategic options - retain, replace, refactor, and rehost - and argues that rehosting through recompilation offers the most favorable balance of cost, risk, and continuity. It then sets out a defensible business case, a phased roadmap that delivers value early, and the governance controls that let modernization be planned rather than gambled.

**Keywords:** *IBM i, AS/400, RPG, COBOL, modernization, rehosting, cloud migration, total cost of ownership*

## **1. The CIO's Dilemma**

Few technology decisions are as quietly consequential as what to do with an IBM i estate. The applications written in RPG and COBOL typically encode decades of refined, market-specific business rules: how an order is priced, how inventory is allocated, how a claim is adjudicated. They run reliably and are, in the truest sense, the system of record. Yet the platform that hosts them is increasingly out of step with the rest of the enterprise, which has standardized on Linux, commodity x86 compute, and the public cloud.

The dilemma is that the perceived alternatives are both unattractive. Doing nothing defers a growing cost and risk; rewriting from scratch invites a multi-year program with full regression of rules no one fully remembers writing. This guide exists because there is a third path that most boards are never shown.

## **2. The Real Cost of Staying on Power**

The cost of remaining on IBM Power is rarely a single line item, which is precisely why it escapes scrutiny. It accumulates across hardware refresh cycles, operating-system and middleware licensing, specialized administration, and the rising premium commanded by a shrinking pool of RPG and IBM i operations talent. As experienced staff retire, the labor market tightens further, and key-person risk concentrates in a handful of individuals.

There is also an opportunity cost. An estate that sits outside the cloud cannot easily share the identity, observability, backup, and automation tooling the rest of the organization has standardized on. Each integration becomes bespoke, and the IBM i island remains an exception in every audit, every disaster-recovery test, and

every security review.

### 3. The Four Options

Modernization strategies for an IBM i portfolio reduce to four archetypes. They are not mutually exclusive, and a mature program usually sequences them.

#### 3.1 Retain

Continue on Power, optionally moving to IBM Power Virtual Server to shed hardware ownership. This is the lowest-effort option and preserves everything, but it also preserves the proprietary dependency, its pricing, and its skills market, and it leaves the estate outside the hyperscale cloud.

#### 3.2 Replace

Adopt packaged software - a standard ERP, for example - in place of custom applications. Where a commercial product genuinely covers the function, replacement can be the right answer. Automotive, distribution, and insurance IBM i estates, however, typically encode plant- and market-specific processes that packages fit poorly without heavy and expensive customization.

#### 3.3 Refactor or Rewrite

Re-engineer the applications into Java, .NET, or cloud-native services. This produces the most modern result but at the highest cost and risk: multi-year timelines, full regression of decades of embedded logic, and dual-running costs throughout. Industry experience suggests rewrites are best applied selectively, after a stabilizing rehost, rather than as the migration vehicle itself.

#### 3.4 Rehost via Recompile

Recompile the unchanged RPG, COBOL, CL, and DDS source to native x86 object code and reproduce the operating environment and DB2/400 database on Linux, enabling deployment on AWS, Azure, or Google Cloud. Business logic, screens, and data structures are preserved; the proprietary hardware dependency is removed. This is the approach taken by Infinite i, and the focus of the remainder of this guide.

## 4. The Rehosting Approach with Infinite i

Infinite i is a compiler and runtime environment that recompiles unaltered IBM i source to native x86 object code and replicates the DB2/400 database on an industry-standard relational engine. Because the applications are recompiled rather than emulated, they run as ordinary Linux workloads, with no specialty appliance and no per-core Power license. Engagements are typically completed in approximately 45 to 60 days, with duration governed chiefly by portfolio size and test scope.

Crucially for the CIO, the result is operationally familiar. The migrated estate deploys with the organization's existing infrastructure-as-code, monitoring, backup, and identity tooling, and the data lands in a standard relational database reachable through ODBC and JDBC. The IBM i island becomes, at last, just another set of Linux services.

## 5. Building the Business Case

A credible business case rests on three numbers the board already asks about: cost, risk, and time. On cost, rehosting replaces open-ended Power refresh and licensing with a one-time, budgetable project and a materially lower run rate on commodity cloud compute. On risk, preserving the compiled business logic removes the single largest hazard of a rewrite - the silent loss of undocumented rules. On time, a 45-to-60-day rehosting window is measured against the multi-year horizon of re-engineering.

The honest caveat is that performance after rehosting is workload-dependent. Prospective adopters should insist on a benchmark of their own workloads rather than rely on generic claims; the structural factors - native object code on modern x86 - favor the target environment, but the proof should be specific to the estate in question.

## 6. A Phased Roadmap

The lowest-risk programs deliver value in stages rather than as a single cutover.

**Phase 1 - Interface.** Deploy a modern browser interface over the existing green screens (for example, with Infinite Cloud), delivering immediate user-facing value with no platform change.

**Phase 2 - Platform.** Run a compatibility scan, confirm the migration timeline and a fixed price, then recompile to native Linux on the chosen cloud. Users keep working; the change happens beneath them.

**Phase 3 - Data.** Migrate the DB2/400 data to a modern relational database (for example, with Infinite DataLink to MySQL), and decommission Power.

## 7. Risk Management and Governance

Modernization should be auditable. Fixed scope and fixed price convert an open-ended liability into a planned project. Validated cutover - row counts and checksums verified end to end - provides the evidence a risk committee requires. Data residency is addressed by deploying within an in-country cloud region where regulation demands it. And because the rehost is reversible in its early phases, the program never reaches a point of no return without explicit sign-off.

## 8. Conclusion

The applications are not the problem; the platform lock-in is. A CIO does not have to choose between an expensive status quo and a risky rewrite. Rehosting through recompilation preserves the value encoded in decades of RPG and COBOL while moving it onto modern, affordable, and operationally familiar infrastructure - a move that can be planned, priced, and defended to the board.

## References

- [1] Infinite Corporation. *Infinite i: Rehosting IBM i Applications to Linux and Public Cloud*. Product documentation, 2026.
- [2] IBM. *IBM i Strategy and Roadmap*. IBM Corporation, 2025.
- [3] Acacio, B. *Migrating IBM i Applications to Public Cloud Without Code Rewrite*. Infinite Corporation, June 2026.
- [4] Amazon Web Services. *Migration Strategies: the 7 Rs*. AWS Prescriptive Guidance, 2025.
- [5] Gartner. *Modernizing Legacy Applications: Balancing Cost, Risk, and Value*. Research note, 2024.