Salem Automation
Systems Integration Specialists

WHITE PAPER · OPENVMS MODERNIZATION

Bridging the Itanium Gap

A practical roadmap for moving mission-critical OpenVMS systems off end-of-life Itanium hardware — without breaking what already works.

KEY TAKEAWAY

Itanium is end-of-life. OpenVMS is not. Salem Automation moves your OpenVMS systems onto modern, supported hardware — preserving the applications, data, and reliability your business runs on.

PREPARED BY
Salem Automation, Inc.
Systems Integration Specialists

DATE
April 2026

AUDIENCE
IT executives & technical leadership

EXECUTIVE SUMMARY

The Itanium Gap

Across manufacturing, utilities, healthcare, defense, and financial services, some of the most critical business systems in operation today run on **OpenVMS** — an operating system prized for decades of uninterrupted reliability. For most of that history, OpenVMS ran on Intel's **Itanium** processor. That foundation has now been withdrawn.

Intel ended Itanium production in 2021, and Hewlett Packard Enterprise ended support for Itanium-based servers at the close of 2025. No new hardware is being built, no security patches are being issued, and the specialists who maintain these systems are retiring. The result is a widening gap between systems that still perform flawlessly and the unsupported hardware beneath them.

Salem Automation closes that gap. We move OpenVMS workloads off Itanium onto modern, fully supported x86-64 servers — preserving the applications, data, and operational behavior your business depends on, with none of the reliability that made OpenVMS the right choice in the first place lost along the way.

SECTION 01

Why Standing Still Is No Longer an Option

Systems running on Itanium still work today. The risk is not that they stop on their own — it is that the ground beneath them has shifted. Five pressures make continued reliance on Itanium untenable.

01

Spare parts are disappearing

When a motherboard or controller fails, replacements may take months to source — if they can be found at all. A single failure can idle production for weeks.

02

Security gaps go unpatched

With no vendor issuing patches, every newly discovered vulnerability stays open indefinitely — a compounding liability for regulated operations.

03

Auditors are flagging it

Under SOX, HIPAA, CMMC, and FDA oversight, end-of-life hardware is increasingly cited as a material risk finding — a problem in its own right, independent of any failure.

04

Expertise is retiring

The pool of engineers who can maintain Itanium-era OpenVMS shrinks every year, and no new specialists are entering it.

05

Budget is trapped

Every dollar spent keeping unsupported hardware alive is a dollar not invested in modern capability. These organizations are maintaining the past instead of building the future.

SECTION 02

Three Solutions That Work Together

Salem addresses the Itanium gap with three coordinated services. Together they cover the full legacy OpenVMS estate — current Itanium workloads and the older Alpha and VAX systems that so often sit alongside them.

1

SOLUTION 1 · THE FOUNDATION

Native OpenVMS on x86-64

The company that now develops and supports OpenVMS has re-engineered the operating system to run natively on standard x86-64 servers: the same Intel Xeon and AMD EPYC platforms that run modern Windows and Linux estates. This gives Itanium customers a supported, long-lived home for their OpenVMS applications.

The move is more than a file copy. Applications must be recompiled from source with tools that target the new architecture, then tested thoroughly to confirm identical behavior. Salem plans the migration, recompiles the applications, validates the result, and manages the cutover — a substantial project that demands exactly the deep OpenVMS expertise we have built our practice around.

WHAT CHANGES

- All application code is recompiled for the x86-64 architecture.
- Some third-party layered products may not yet be available on the new platform.
- Any Itanium-specific machine code must be rewritten.
- Clusters cannot mix Itanium and x86 nodes, so cutover must be coordinated.

WHAT STAYS THE SAME

- The OpenVMS operating system — same commands, tools, and security model.
- The file system and how applications store and retrieve data.
- Network configuration and communication protocols.
- The fundamental reliability and behavior that made OpenVMS valuable.

2

SOLUTION 2 · THE ENGINEERING

Application Porting & Revalidation

Porting is the engineering work behind the move: taking software built for one architecture and making it run correctly on another. It is where the majority of project effort lives. Salem's porting work runs in four disciplines:

Source-code review

We examine the application code to find everything written specifically for Itanium hardware, then update or rewrite it.

Rebuild

We stand up the new compiler toolchain and rebuild every application to run on the new chips.

Comprehensive testing

Unit, integration, performance, stress, and user-acceptance testing confirm the system behaves exactly as before.

Build-system updates

The automated pipelines that compile and package the software are updated for the new platform.

Note — This is not “just a recompile.” Every application must be revalidated, and many organizations have run stable for decades without a maintained test suite. Building that test coverage is often part of the engagement — and it becomes a durable quality asset long after the migration is complete.

3

SOLUTION 3 · THE OLDER ESTATE

Emulation for Alpha & VAX

Many organizations that run OpenVMS on Itanium also operate older systems on **DEC Alpha** (1990s) and **VAX** (1980s) hardware — systems that were never moved to Itanium and have quietly run for decades. Because their source code was never brought forward, these applications generally cannot be recompiled. **Emulation** solves this: software makes a modern server behave exactly like the original hardware, so the applications run unchanged.

vtAlpha

Presents a modern server as DEC Alpha hardware. Runs OpenVMS Alpha applications with no changes to the application code.

vtVAX

Presents a modern server as VAX hardware. Runs OpenVMS VAX applications — some more than 30 years old — with no changes.

Emulation does not replace the x86-64 port; the native x86-64 port is the answer for Itanium workloads. But together they let Salem address the *entire* legacy estate — native x86-64 for Itanium, emulation for Alpha and VAX — all on modern, supported servers.

SECTION 03

The Database Challenge

Moving the operating system and applications is one half of the picture. Most OpenVMS environments also run databases — and many of those databases are themselves old, unsupported, or tied to the legacy platform.

Migration is rarely a simple data transfer. These systems hold decades of embedded business logic — formulas, validations, and automated processes written for the original platform — all of which must be translated, tested, and proven on the new database. The table below maps each common OpenVMS database to its realistic modern targets.

Oracle Rdb

The most common OpenVMS database, purpose-built for the platform, with a great deal of business logic embedded inside it.

MIGRATES TO

PostgreSQL | Oracle on Linux | SQL Server

Sybase

A commercial database that shares lineage and a similar query language with Microsoft SQL Server.

MIGRATES TO

SQL Server · easiest | PostgreSQL

Progress OpenEdge

Tightly coupled to the Progress programming language, which makes the database hard to separate from the application.

MIGRATES TO

PostgreSQL | SQL Server | Progress on Linux

HP DBMS (CODASYL)

HARDEST MIGRATION

A network-model database that organizes data very differently from modern relational systems — connections rather than tables and rows.

MIGRATES TO

PostgreSQL · full redesign

RMS Indexed Files

Not a database, but used as one: the OpenVMS file system itself, with built-in keys and indexing, as an application data store.

MIGRATES TO

PostgreSQL · rewrite data access

FOR TECHNICAL TEAMS

The difficulty concentrates in **stored procedures** — the rules, calculations, and validations that live inside the database rather than the application. Each must be translated to the target platform's procedural dialect and validated against known results from the source system before cutover.

SECTION 04

A Proven Four-Phase Migration Method

Salem follows the same disciplined method for every database migration — a sequence designed to eliminate surprises and prove correctness at each step.

PHASE 1

Discovery

Catalog the environment — data, structure, embedded logic, connected systems, and current performance — producing a clear scope and difficulty assessment.

PHASE 2

Design

Decide the target platform, how the schema must change, and how business logic will be translated — worked out with your database and application teams.

PHASE 3

Convert & Test

Convert the schema, translate the logic, move the data, and test extensively — validating every rule against known results from the source system.

PHASE 4

Go Live

Cut production over with a detailed runbook, a tested rollback path, and 30 days of intensive post-cutover support to catch and fix any issue.

SECTION 05

The Strategic Path: Stabilize First, Then Migrate

The most important strategic principle is this: **you do not have to do everything at once.** Salem recommends a phased path that removes the most urgent risk first, then modernizes on your timeline. Each stage is a decision point — not a mandate.

1

Stabilize

Move OpenVMS off Itanium onto modern servers — native x86-64 for Itanium workloads, emulation for Alpha and VAX.

WHY IT MATTERS Stops the bleeding. Eliminates the risk of a hardware failure taking down production.

2

Assess

Map the database landscape — what must move, how difficult it will be, and what it will cost.

WHY IT MATTERS No surprises. Leadership gets an honest picture before committing to migration.

3

Migrate

Move databases to modern platforms, translate business logic, and test thoroughly.

WHY IT MATTERS Modern, supported databases. Lower licensing costs. Connectivity to current tools and APIs.

4

Optimize

Modernize the applications themselves — build APIs, evaluate cloud, and decommission legacy infrastructure.

WHY IT MATTERS Full modernization — on current technology, with room to grow.

> You do not have to choose between keeping operations running and modernizing. Stage 1 buys time and removes the most urgent risk immediately. From there, databases and applications can be migrated at your own pace — without failing hardware forcing the schedule.

SECTION 06

The Compliance Imperative

For many organizations, compliance is the decisive factor. Running end-of-life hardware with no security patching is a compliance problem in nearly every regulated industry, and auditors increasingly cite unsupported infrastructure as a material risk finding. Moving to supported platforms — both hardware and databases — directly resolves those findings.

NIST 800-171
Controlled unclassified information

IEC 62443
Industrial & OT security

SOX
Financial reporting controls

HIPAA
Healthcare data protection

FDA 21 CFR Part 11
Pharmaceutical manufacturing records

The practical consequences of inaction are real: failed audits, conditional findings, and remediation mandates. For regulated buyers, modernization is not only technically sound — it is compliance-necessary.

SECTION 07 · APPENDIX

Glossary of Key Terms

Itanium / IA-64
Intel's now-discontinued server chip — the hardware customers need to move off of.

OpenVMS
A highly reliable operating system that runs critical business applications, many for decades.

Recompile
Rebuilding an application from its source code so it runs on a different type of chip.

Oracle Rdb
The most common OpenVMS database — a distinct product from Oracle Database.

PostgreSQL
A modern, open-source database that is the default migration target for most legacy systems.

CODASYL
A network-model database design (used by HP DBMS) — the hardest type to migrate.

x86-64
The standard chip architecture in virtually all modern servers (Intel Xeon, AMD EPYC).

Cluster
A group of OpenVMS servers working together as one system. Itanium and x86-64 nodes cannot be mixed in a single cluster.

Emulation
Software that makes a modern computer behave like old hardware, so programs run unchanged.

Stored procedures
Business logic that lives inside the database itself — what makes migrations hard.

RMS
Record Management Services — the OpenVMS file system's built-in keyed data storage.

Layered products
Third-party software that runs on top of OpenVMS — not all are ported to x86-64 yet.

Have questions about your OpenVMS environment?

Salem Automation brings decades of hands-on OpenVMS expertise to every stage of this journey — from stabilizing Itanium hardware today to migrating databases and modernizing applications on your timeline. We would welcome the chance to map the right path for your systems.

CONTACT US
sales@salemautomation.com
Tel: +1 336-661-0890

Salem Automation, Inc.
3909 Westpoint Blvd. · Suite C
Winston-Salem, NC 27103
www.salemautomation.com