

CBA Salesforce SAP and AWS in 2026. Is there still some use of cobol? 2:36 pm 28 Jul 2026

Yes, **COBOL is still actively used in 2026**, both globally and even contextually within the modernised architectures of major institutions like the **Commonwealth Bank of Australia (CBA)**. While modern frontends and cloud platforms hog the spotlight, COBOL remains the silent, load-bearing infrastructure of global finance.

The Global Reality of COBOL in 2026

Despite decades of predictions about its demise, COBOL remains a dominant force in corporate America and global banking:

- **The Numbers:** An estimated **800 billion to 1 trillion lines of COBOL code** are actively running in production worldwide as of 2026.
- **The Scale:** COBOL still powers close to **90% of Fortune 500 business systems** and processes roughly **85% of all daily business transactions**, including 95% of all ATM swipes.
- **The "Money Math" Advantage:** COBOL is surviving because it handles exact decimal arithmetic by default. Unlike modern languages that rely on binary floating-point math (where minor rounding errors can happen), COBOL was hardcoded from the ground up to never introduce a "penny-drift" bug across millions of sequential records.

Where COBOL Fits alongside Salesforce, SAP, and AWS

When major banks migrate to modern toolstacks like **Salesforce (for CRM)**, **SAP (for core transactional logic)**, and **AWS (for cloud infrastructure)**, COBOL does not completely vanish overnight. Instead, it undergoes a tactical shift:

1. The Phased Cloud Migration (e.g., CBA)

CBA represents a masterclass in modern core banking transformation. In late 2025, **CBA completed a massive migration**, moving its SAP-based core platforms for retail and institutional banking into **AWS**. This unlocked 90% of its data for cloud-native AI and digital tools.

However, even a tech-forward giant like CBA moves in phases. While their primary deposit and transaction ledger runs on modern cloud-hosted SAP S/4HANA infrastructure, the bank is still systematically migrating its final legacy pipelines—such as credit card management—into AWS throughout 2026. Many of these lagging secondary systems or peripheral settlement lines globally are still tethered to COBOL engines or Mainframe I/O dependencies.

2. Hybrid "Mainframe Modernization" on AWS

AWS doesn't always mean "destroying COBOL." Instead of doing high-risk, multi-billion-dollar rewrites, many enterprises utilize services like **AWS Mainframe Modernization**. These tools use advanced compilers (like Rocket Visual COBOL or AWS Blu Age) to take existing COBOL code, containerise it, and run it directly inside AWS x86 cloud architecture with minimal changes.

3. Wrapped Microservices

Many companies choose to leave their foundational COBOL batch-processing logic exactly where it is. They use APIs to create a middleware layer, allowing **Salesforce** cloud applications to securely pull and push data to a backend mainframe that still talks in COBOL.

4. The 2026 AI Inflection Point

The reason you see rapid shifts toward cloud architectures in 2026 is thanks to **Generative AI**. Tooling has advanced to the point where AI-driven automated code refactoring can now convert COBOL to Java with **70–85% accuracy** (up from just 40% a few years ago). This has drastically dropped the cost and risk of legacy migrations.

**** End of Page**