OpenText Alchemy Stabilization and Upgrade

Invoice and Order Tracking System

Executive Summary

- **Project Name**: OpenText Alchemy Stabilization and Upgrade
- **Project Duration**: December 2–4, 2024 (3 days)
- **Project Owner/Author**: Charlie Hofner
- **LinkedIn**: LinkedIn
- **Project Sponsor/Stakeholder**: Finance/Operations Director, Accounts Payable Team

Overview
Stabilized a legacy OpenText Alchemy (Document Server, Alchemy Edition) installation used at one location for centralized tracking of invoice PDFs linked to sales/purchase orders and warranty status records.

The **primary driver** for the project was the discovery that the system was running a highly vulnerable and unsupported version of SQL Server 2012 on Windows Server 2022. Vendor support provided no useful guidance on compatibility.

Over three days in early December 2024, performed a stepwise SQL backend upgrade (2012 → 2016 → 2019) to eliminate the critical SQL vulnerabilities. As a **bonus** during the process, successfully upgraded the Alchemy application from a very outdated version 7.0 to version 9.0 — the latest and final major release.

All changes were executed with staff coordination to avoid disruption and protected by VMware snapshots. The upgrades bought significant time for the business to evaluate and select a modern replacement while maintaining secure, reliable operation.

Key Outcomes - Eliminated critical vulnerabilities in SQL Server 2012 - SQL backend modernized to fully supported SQL Server 2019 - Alchemy upgraded to final version 9.0 as an unexpected bonus - Extended usable life of the system with minimal disruption - Zero downtime during business hours

Background and Objectives

Business Need

The Alchemy system was critical for invoice, order, and warranty tracking but was running on a severely vulnerable and unsupported SQL Server 2012

installation. This posed an unacceptable security risk. The business needed the system to remain operational while a long-term replacement was identified and implemented.

Objectives

- **Primary**: Upgrade SQL backend to a supported, patched version to remediate vulnerabilities
- **Secondary**: Maintain full Alchemy functionality with no business disruption
- **Bonus opportunity**: Upgrade Alchemy application if feasible during the process
- Buy time for proper evaluation and migration to a modern ECM solution

System Overview

Technical Environment & Upgrades

- **Host OS**: Windows Server 2022 (VMware virtual machine)
- **Alchemy Version**:
  - Before: 7.0 (very outdated)
  - After: 9.0 (latest and final major release, ~2010–2012) — upgraded as a bonus
- **Database Backend**:
  - Before: SQL Server 2012 (highly vulnerable and unsupported)
  - After: SQL Server 2019
- **Safety Measures**: All changes protected by VMware snapshots for instant rollback

Project Timeline & Upgrade Execution

Monday, December 2, 2024
- Project start (driven by SQL vulnerability remediation) - Afternoon (after staff finished using system and gave go-ahead): - Performed stepwise SQL upgrade: 2012 → 2016 → 2019 - Restarted server after each step - Verified SQL services running - Tested database connection and Alchemy Administration Console — successful - Left system for staff testing the following day

Tuesday, December 3, 2024
- Morning/afternoon: Staff confirmed all functions working normally - Afternoon (with SQL now secure and stable): - Took opportunity to upgrade Alchemy application: 7.0 → 9.0 - Restarted server - Personally verified full functionality post-restart

Wednesday, December 4, 2024
- Morning: Staff performed final checks and confirmed everything operating properly - Project completed — system secure and running latest possible Alchemy version

Specific Use Case: Invoice PDF Tracking with Orders and Warranty Status

The system served as a central archive linking: - Invoice PDFs (indexed by Invoice #, Vendor, Amount, PO #) - Purchase/Sales Orders - Warranty Records (with Start/End dates and Status)

Document Type	Key Indexing Fields	Example Usage	Retention Policy Example
Invoice PDFs	Invoice #, Date, Vendor, Amount, PO #	AP processing, disputes	7 years post-payment
Purchase/ Sales Orders	Order #, Customer, Date, Items	Full transaction audit trail	10 years
Warranty Records	Serial #, Warranty Start/End, Status	Claims/returns authorization	Until expiration + 2 years

Methodology and Approach

- Primary focus: Remediate vulnerable SQL 2012 installation
- Coordinated timing with end users to eliminate business impact
- Used VMware snapshots before each major change
- Stepwise SQL upgrades with immediate verification
- Opportunistic Alchemy upgrade once SQL stability was confirmed
- Multi-stage user acceptance testing (Charlie → staff)

Challenges and Solutions

- **Challenge**: Critical vulnerabilities in unsupported SQL Server 2012
  Solution: Stepwise upgrade to SQL Server 2019 with snapshot protection and thorough verification
- **Challenge**: Unhelpful vendor support on SQL/Alchemy compatibility
  Solution: Relied on internal expertise, testing, and rollback safety nets
- **Challenge**: Minimizing disruption to active users
  Solution: Afternoon/evening scheduling, staff coordination, staged verification

Lessons Learned

- Addressing the root security issue (SQL) can create safe opportunities for additional improvements (Alchemy upgrade)
- Snapshots enable confident action on legacy systems
- Clear communication with users builds trust and enables quick turnaround

Results and Metrics

- **Duration**: 3 business days (Dec 2–4, 2024)
- **Downtime**: None during business hours
- **Success Rate**: 100% — SQL vulnerabilities eliminated, Alchemy upgraded as bonus
- **Outcome**: System now secure and running the latest possible version of Alchemy, providing valuable breathing room for replacement planning
- **Next Steps**: Evaluate and implement modern ECM replacement (e.g., OpenText Content Suite/Cloud or alternative) to fully retire legacy Alchemy

Appendices

Upgrade Safety Checklist (Used Throughout)

1. Coordinate timing with staff
2. Take descriptive VMware snapshot
3. Perform upgrade step
4. Restart server
5. Verify SQL services
6. Test Alchemy Admin Console login and core functions
7. Staff verification
8. Remove snapshot