



Datenmigration auf neuem Level - mit Eigenentwicklung & Audit-Trail

Datenmigration – komplex, risikobehaftet, erfolgskritisch. Gerade bei ERP-Systemen wie NetSuite entscheidet die Qualität des Migrationsprozesses über Projekterfolg oder Chaos.

In diesem Projekt wurde eine performante, auditierbare und modulare Lösung umgesetzt:

- Eigene ETL-Entwicklung in C# (.NET)
- Staging über einen MS-SQL Data-Lake
- Vollständiges Audit-Konzept inkl. Änderungsverfolgung
- Automatisierte Reports & Cleansing
- API-Integration zu NetSuite via SuiteTalk & SuiteAnalytics

Das Ergebnis: Höchste Transparenz, flexible Wiederholbarkeit, minimaler manueller Aufwand – und ein zufriedener Kunde.

Zielsetzung

Effiziente und nachvollziehbare Datenmigration nach NetSuite.

- Übertragung von Stamm- und Bewegungsdaten aus diversen Quellsystemen
- Nutzung eines eigens entwickelten C#-ETL-Tools
- Vollständige Audit-Dokumentation
- Einsatz eines MS-SQL-Servers als Data-Lake/Staging-Umgebung

Voraussetzungen

Was musste vorhanden sein?

- Quellsystem-Inventarisierung: SAP, Dynamics, Infor, proprietär
- Technische Anbindung: ODBC/JDBC, Web-APIs
- Zielsystem: NetSuite (SOAP/REST APIs, SuiteAnalytics Connect)
- C#-Entwicklungsumgebung (Visual Studio, .NET 7)
- MS-SQL-Server als Data-Lake
- Durchdachtes Audit-Konzept (Log-Datenbank oder Files)

Prozessablauf (Visualisierung als Flowchart)

Vom Alt-System ins Zielsystem – mit Kontrolle in jeder Phase

- Analyse & Mapping
- Datenextraktion → MS-SQL-Server Staging
- Transformation mit Audit-Trail
- Validierung & Cleansing
- Laden in NetSuite
- Post-Load-Prüfung
- Go-Live & Monitoring

Datenverarbeitung im Detail

Schrittweise Umsetzung mit Fokus auf Qualität.

- Mapping & Geschäftsregeln (Feld-zu-Feld, Typkonvertierung)
- Extraktion via SQL, APIs oder Flat-Files (Voll- & Inkrement-Export)
- Staging mit BatchID & Zeitstempel
- Transformation & Audit durch C#-Service inkl. Fehlertracking und Fehlerhandling
- Validierung: Pflichtfelder, Formate, Duplikate (Fuzzy-Matching)

Ladeprozess & Auditing

Sicherheit & Transparenz beim Load nach NetSuite.

- Upload über SuiteTalk oder SuiteAnalytics CSV
- Lade-Reihenfolge beachten: Stammdaten vor Bewegungsdaten
- Jede Änderung protokolliert (UserID, Zeitstempel, Alt → Neu)
- Automatisierte Reports und Datenabgleiche

Rollen & Tools

Zusammenspiel von Experten & Technologien

- Projektleitung: Zeitplan, Kommunikation, Ressourcen
- Datenarchitekt: Zielmodell, Schnittstellen, Audit-Design
- Entwickler: C#-ETL-Tool für E-T-L-A
- DBA: MS-SQL-Server & Oracle-Performance, Backups
- Fachbereich: Datenvalidierung, Tests, Freigabe

Technologien

Was kam zum Einsatz?

- C# (.NET 7), MS-SQL-Server (2019/2022), Oracle/NetSuite
- SuiteTalk SOAP/REST, SuiteAnalytics Connect
- EF Core Logging für Audit

Best Practices & Lessons Learned

Was den Unterschied gemacht hat.

- Modularer Aufbau des ETL-Tools
- Konfigurierbare Mapping-Files (JSON/XML)
- Iterative Mock-Testläufe
- Rollback-Optionen je Ladephase
- Durchgängige Dokumentation & Versionierung
- Fachbereich frühzeitig einbinden

Kontakt

Kommen wir ins Gespräch

Gemeinsam entwickeln wir die Best-Practice-Lösungen, die perfekt zu Ihren Zielen passen.

Sie erreichen uns unter:



+49 2666 91174-50



vertrieb@aponsi.com