



# Windows Server 2003 → 2008

## Migration Analysis

---

### KIS Banking Application - Platform Migration Assessment

---

**Datum analýzy:** 2025-12-01 **Analyzováno:** 1,021 Java tříd + 1,288 JSP stránek **Databáze:** Neo4J (grafy)  
+ Qdrant (264,740 code chunks)

---



### Executive Summary

---

#### Migrace

- **Zdrojová platforma:** Windows Server 2003 (32-bit)
- **Cílová platforma:** Windows Server 2008 (32-bit)
- **Java verze:** Java 1.4 (originální verze aplikace)
- **Architektura:** 32-bit → 32-bit (beze změny)

#### Risk Assessment

**Risk Level:** ⚠ STŘEDNÍ **Verdict:** ⚠ MIGROVATELNÉ S ÚPRAVAMI

*Některé oblasti vyžadují úpravu kvůli User Account Control (UAC) a deprecated Java 1.4 API.*

#### Klíčové Statistiky

- **Celkem tříd:** 1,021
  - **Celkem problémů:** 6,936
  - **Kritické soubory:** 1 (Java 1.4 deprecated API)
  - **Vysoké riziko soubory:** 81 (Protected file system)
  - **Complexity Score:** 80/200
-

## ⚠ Detailní Breakdown Problémů

### 1. Java 1.4 Deprecated APIs ❌ VYSOKÉ RIZIKO

Nalezeno: 6 výskytů v 1 souboru

Problémový soubor: - `/sources/JSP/idm.jsp`

Deprecated API:

```
sun.misc.BASE64Encoder
sun.misc.BASE64Decoder
```

#### Problém

Java 1.4 používá deprecated `sun.misc.BASE64Encoder/Decoder`, které: - ❌ Byly odstraněny v Java 9+ - ⚠ Nefungují reliable v Java 7-8 - ⚠ Nejsou součástí standardního JDK API

#### Řešení

PŘED (Java 1.4 - deprecated):

```
<%@ page import="sun.misc.BASE64Encoder" %>
<%
    String data = "Hello World";
    BASE64Encoder encoder = new BASE64Encoder();
    String encoded = encoder.encode(data.getBytes());
    out.println("Encoded: " + encoded);
%>
```

PO (Java 1.6+ - standard API):

```
<%@ page import="javax.xml.bind.DatatypeConverter" %>
<%
    String data = "Hello World";
    String encoded = DatatypeConverter.printBase64Binary(data.getBytes("UTF-8"));
    out.println("Encoded: " + encoded);
%>
```

PO (Java 8+ - java.util.Base64):

```
<%@ page import="java.util.Base64" %>
<%
    String data = "Hello World";
    String encoded = Base64.getEncoder().encodeToString(data.getBytes("UTF-8"));
    out.println("Encoded: " + encoded);
%>
```

## Doporučení

1. **Immediate:** Nahradit `sun.misc.BASE64Encoder` za `javax.xml.bind.DatatypeConverter` (Java 1.6+)
2. **Long-term:** Upgrade na `java.util.Base64` (Java 8+)
3. **Testing:** Unit testy pro encoding/decoding kompatibilitu

## 2. Protected File System (UAC Virtualization) ⚠ STŘEDNÍ RIZIKO

**Nalezeno:** 405 výskytů v 81 souborech

**Problémové oblasti:** - `C:\Windows\` (chráněno UAC) - `C:\Program Files\` (chráněno UAC) - Hardcoded paths k system directories

### Problém

Windows Server 2008 zavedl **User Account Control (UAC)**, který virtualizuje přístup k: - `C:\Windows\` - `C:\Program Files\` - `HKEY_LOCAL_MACHINE\SOFTWARE`

**Důsledky:** - ⚠ Aplikace bez admin práv zapíše do virtualizované lokace - ⚠ Soubory se zapíší do `%LOCALAPPDATA%\VirtualStore\` - ⚠ Jiný uživatel neuvidí změny

### Příklad problémového kódu

**PŘED (Windows 2003 - bez UAC):**

```
// ESEExportMustky.java
public class ESEExportMustky {
    private static final String EXPORT_PATH = "C:\\Windows\\Temp\\exports\\";

    public void exportData(List<Data> data) throws IOException {
        File exportDir = new File(EXPORT_PATH);
        if (!exportDir.exists()) {
            exportDir.mkdirs(); // ❌ Selže na Win 2008 bez admin práv
        }

        File exportFile = new File(EXPORT_PATH + "export.xlsx");
        // ... export logic
    }
}
```

### Typické UAC Virtualization Flow:

```
Aplikace:   Zapsat do C:\Windows\Temp\exports\export.xlsx
UAC:        Aplikace nemá admin práva
UAC:        → Virtualizuj do %LOCALAPPDATA%\VirtualStore\Windows\Temp\exports\
Výsledek:   Soubor je v uživatelském profilu, ne v systémové složce
```

## Řešení

### PO (Windows 2008 - UAC compliant):

```
public class ESExportMustky {
    // ✅ Použij %APPDATA% nebo %ProgramData%
    private static final String EXPORT_PATH =
        System.getenv("PROGRAMDATA") + "\\KIS\\exports\\";

    // Nebo:
    // System.getenv("APPDATA") + "\\KIS\\exports\\"

    public void exportData(List<Data> data) throws IOException {
        File exportDir = new File(EXPORT_PATH);
        if (!exportDir.exists()) {
            exportDir.mkdirs(); // ✅ Funguje bez admin práv
        }

        File exportFile = new File(EXPORT_PATH + "export.xlsx");
        // ... export logic
    }
}
```

## Migrační strategie

### Fáze 1: Identifikace (1 týden)

```
# Najít všechny hardcoded Windows paths
grep -r "C:\\\\Windows" sources/JAVA/
grep -r "C:\\\\Program Files" sources/JAVA/
```

**Fáze 2: Refactoring (3 týdny)** - Nahradit hardcoded paths za environment variables - Centralizovat path configuration do `config.properties` - Vytvořit `PathManager` utility class

### Fáze 3: Testing (2 týdny)

```
// Test s UAC zapnutým
@Test
public void testExportWithUAC() {
    // Simuluj non-admin uživatele
    ESExportMustky export = new ESExportMustky();
    export.exportData(testData);

    // Verify file je ve správné lokaci
    String expected = System.getenv("PROGRAMDATA") + "\\KIS\\exports\\export.xlsx";
    assertTrue(new File(expected).exists());
}
```

### Doporučené lokace pro Windows 2008:

Typ dat	Windows 2003	Windows 2008	UAC Safe
App Config	C:\Program Files\KIS\config\	%PROGRAMDATA%\KIS\config\	✓
User Data	C:\Windows\Temp\kis_user\	%APPDATA%\KIS\	✓
Shared Data	C:\Windows\Shared\kis\	%PUBLIC%\KIS\	✓
Temp Files	C:\Windows\Temp\	%TEMP%\	✓
Logs	C:\Program Files\KIS\logs\	%PROGRAMDATA%\KIS\logs\	✓

### 3. File Encoding (windows-1250) ✓ NÍZKÉ RIZIKO

Nalezeno: 6,525 výskytů v 644 JSP souborech

#### Analýza

- **Encoding:** windows-1250 (Středoevropské jazyky - čeština)
- **Risk Level:** ✓ NÍZKÉ
- **Kompatibilita:** Windows 2008 plně podporuje windows-1250

#### Doporučení

✓ **ZACHOVAT windows-1250** pro české znaky:

```
<%@ page contentType="text/html; charset=windows-1250" %>
```

**Důvody:** 1. ✓ Windows 2008 má plnou podporu windows-1250 2. ✓ Oracle databáze používá win-1250 (NLS\_CHARACTERSET) 3. ✓ Žádné problémy s českou diakritikou 4. ⚠ Migrace na UTF-8 by vyžadovala: - Database conversion (COSTLY) - Všechny JSP přepsat (644 souborů) - Re-testing celé aplikace

**Pouze pokud migrujete na cloud nebo Linux v budoucnu:** - Zvažte UTF-8 conversion jako separátní projekt - Testujte důkladně na dev/stage prostředí



## Migration Timeline & Phases

### Phase 1: Preparation (1 týden)

**Cíle:** - ✓ Audit všech Windows 2003-specific dependencies - ✓ Setup Windows Server 2008 test environment - ✓ Backup současné produkční konfigurace

**Aktivita:** 1. Vytvořit Windows Server 2008 VM 2. Nainstalovat Java 1.4 JDK 3. Setup Oracle database client 4. Přenést aplikaci do test environmentu

**Deliverables:** - Windows 2008 test environment ready - Migracní checklist

## Phase 2: Code Migration (4 týdny)

### Week 1: Java 1.4 Deprecated API Fixes

**File:** `/sources/JSP/idm.jsp`

**Úkoly:** 1. Nahradit `sun.misc.BASE64Encoder` → `javax.xml.bind.DatatypeConverter` 2. Unit testy pro Base64 encoding/decoding 3. Integration test idm.jsp functionality

**Effort:** 2 dev-days

### Week 2-3: Protected File System Refactoring

**81 souborů v** `/sources/JAVA/src/cz/jtbank/konsolidace/excel/`

**Strategie:**

**1. Vytvořit PathManager utility (1 den):**

```
package cz.jtbank.konsolidace.util;

public class PathManager {
    private static final String BASE_DIR = System.getenv("PROGRAMDATA") + "\\KIS\\";

    public static String getExportPath() {
        return BASE_DIR + "exports\\";
    }

    public static String getTempPath() {
        return System.getenv("TEMP") + "\\KIS\\";
    }

    public static String getConfigPath() {
        return BASE_DIR + "config\\";
    }

    public static void ensureDirectoryExists(String path) {
        File dir = new File(path);
        if (!dir.exists()) {
            dir.mkdirs();
        }
    }
}
```

**2. Refactor Excel export classes (8 dnů):**

```
// PŘED
String exportPath = "C:\\Windows\\Temp\\exports\\";

// PO
String exportPath = PathManager.getExportPath();
PathManager.ensureDirectoryExists(exportPath);
```



**Automatizace s regex replace:**

```
# Find & replace pattern
find . -name "*.java" -exec sed -i 's/C:\\\\Windows\\\\Temp/PathManager.getTempPath()/g' {} \;
```

**3. Testing s UAC enabled (3 dny):** - Test každého exportu s non-admin user - Verify file locations - Test read/write permissions

**Effort:** 12 dev-days





**Week 4: Integration Testing**

**Testy:** 1.  Všechny Excel exporty fungují 2.  idm.jsp Base64 encoding/decoding 3.  File permissions s UAC enabled 4.  Multi-user scenarios

**Effort:** 5 dev-days

**Phase 3: UAC Testing & Configuration (2 týdny)**

**Scénáře k testování:**

Scenario	User Type	Expected Behavior
Excel Export	Non-admin	 Export do %PROGRAMDATA%\KIS\exports\
Config Load	Non-admin	 Read z %PROGRAMDATA%\KIS\config\
Temp Files	Non-admin	 Write do %TEMP%\KIS\
Log Files	Non-admin	 Write do %PROGRAMDATA%\KIS\logs\

**UAC Policies:**

```
Computer Configuration
├─ Windows Settings
│   └─ Security Settings
│       └─ Local Policies
│           └─ Security Options
│               └─ User Account Control: Behavior of the elevation prompt for standard users
│                   → Automatically deny elevation requests
```

**Effort:** 10 dev-days

---

## Phase 4: Production Deployment (1 týden)

### Deployment Checklist:

**Pre-deployment:** - [ ] Backup produkční databáze - [ ] Backup aplikačních souborů - [ ] Dokumentace rollback plánu - [ ] Setup monitoring

**Deployment:** 1. **Stop aplikace** (maintenance window) 2. **Deploy nové soubory** 3. **Vytvořit %PROGRAMDATA%\KIS\ strukturu:**

```
%PROGRAMDATA%\KIS\
├─ config\
├─ exports\
├─ logs\
└─ temp\
```

4. **Set permissions** (BUILTIN\Users má Read+Write) 5. **Start aplikace** 6. **Smoke tests**

**Post-deployment Monitoring** (první 48 hodin): - Monitor error logs - Check UAC event logs - Verify export files location - Test s multiple users

**Rollback Plan** (v případě kritického selhání): 1. Stop Win 2008 server 2. Přepnout DNS zpět na Win 2003 3. Start Win 2003 server 4. Root cause analysis

**Effort:** 5 dev-days

---







## Effort & Cost Estimates

### Manual Migration

Phase	Duration	Effort (dev-days)	Cost (€800/day)
Phase 1: Preparation	1 týden	5 days	€4,000
Phase 2: Code Migration	4 týdny	19 days	€15,200
Phase 3: UAC Testing	2 týdny	10 days	€8,000
Phase 4: Deployment	1 týden	5 days	€4,000
TOTAL	8 týdnů	39 dev-days	€31,200

### With AI Assistance (Claude/GitHub Copilot)

AI může pomoci s: 1.  Automatizace Find & Replace patterns 2.  Generování PathManager utility 3.  Unit test generation 4.  Code refactoring suggestions




Úspora: - **Effort reduction:** 30% (39 → 27 dev-days) - **Cost savings:** €9,360 - **Faster delivery:** 8 týdnů → 5.5 týdnů

Approach	Effort	Cost	Duration
Manual	39 dev-days	€31,200	8 týdnů
With AI	27 dev-days	€21,840	5.5 týdnů
Savings	12 days (30%)	€9,360	2.5 týdny

## Risk Mitigation Strategies

### 1. Java 1.4 Deprecated API Risk

**Risk:** Base64 encoding/decoding nefunkční po migraci

**Mitigation:** 1.  Unit testy pro encoding/decoding před migrací 2.  Side-by-side testing (Win 2003 vs 2008) 3.  Rollback plán připraven

## 2. UAC Virtualization Risk

**Risk:** Aplikace zapíše soubory do neočekávané lokace

**Mitigation:** 1. ☒ Comprehensive UAC testing s non-admin users 2. ☒ File system monitoring (Process Monitor) 3. ☒ Centralizovaný PathManager s logging

---

## 3. Multi-User Access Risk

**Risk:** Různí uživatelé nevidí soubory ostatních

**Mitigation:** 1. ☒ Použít %PROGRAMDATA% (shared pro všechny users) 2. ☒ Set správné NTFS permissions (BUILTIN\Users = Read+Write) 3. ☒ Test s multiple concurrent users

---



## Success Metrics

---

### Functional Metrics

- ☒ Všech **1,288 JSP stránek** funguje bez chyb
- ☒ Všech **81 Excel export tříd** exportuje do správné lokace
- ☒ Base64 encoding/decoding v `idm.jsp` funguje správně
- ☒ Multi-user scenarios (min. 5 concurrent users) fungují

### Performance Metrics

- ☒ Response times **<10% pomalejší** než Win 2003
- ☒ Excel export time **<15% pomalejší**
- ☒ Database query time **beze změny**

### UAC Compliance Metrics

- ☒ **0 UAC virtualization** events v event logu
  - ☒ **100%** souborů v expected %PROGRAMDATA% lokacích
  - ☒ **0 permission denied** errors
-

## Recommended Migration Strategy

---

### Doporučený Přístup: Phased Rollout

#### Step 1: Development Environment (Week 1)

- Migrace dev prostředí na Win 2008
- Developer testing
- Rychlé iterace fixes

#### Step 2: QA/Testing Environment (Week 2-5)

- Full QA testing cycle
- UAC compliance testing
- Performance benchmarking

#### Step 3: Staging/Pre-Production (Week 6)

- Mirror produkce na Win 2008
- Production-like load testing
- Final UAC validation

#### Step 4: Production Rollout (Week 7-8)

- Maintenance window deployment
  - Post-deployment monitoring
  - Rollback plan ready
-

## Comparison: Win 2003 vs Win 2008

Feature	Windows 2003	Windows 2008	Impact
UAC	✗ No	✓ Yes	⚠ Requires file path changes
Java 1.4 Support	✓ Full	✓ Full	✓ Compatible
32-bit	✓ Yes	✓ Yes	✓ No changes needed
windows-1250	✓ Yes	✓ Yes	✓ Compatible
Oracle Client	✓ Yes	✓ Yes	✓ Compatible
File Virtualization	✗ No	✓ Yes	⚠ Need UAC testing

## Final Recommendations

### Immediate Actions (Before Migration)

#### 1. Code Changes:

- Replace `sun.misc.BASE64Encoder` in `idm.jsp`
- Refactor 81 Excel export classes to use PathManager
- Create centralized path configuration

#### 2. Testing Setup:

- Windows 2008 test environment
- UAC enabled testing
- Non-admin user scenarios

#### 3. Documentation:

- Migration runbook
- Rollback procedures
- UAC troubleshooting guide

## Post-Migration Actions

### 1. Monitoring (první měsíc):

- UAC event log monitoring
- File system access patterns
- Application error logs

### 2. Performance Tuning:

- Benchmark comparison Win 2003 vs 2008
- Optimize slow operations
- Database query optimization

### 3. User Training:

- UAC behavior changes
- File location changes
- Troubleshooting common issues

---

## Conclusion

Migrace z Windows Server 2003 na Windows Server 2008 je **PROVEDITELNÁ** s těmito podmínkami:

✅ **Low-risk areas:** - Java 1.4 kompatibilita - 32-bit architektura - windows-1250 encoding

⚠️ **Medium-risk areas:** - UAC file virtualization (vyžaduje refactoring) - Protected file system access (81 souborů)

❌ **High-risk areas:** - Deprecated Java 1.4 API (1 soubor - snadné opravit)

**Celkové doporučení:**

*Pokračujte s migrací pomocí **phased rollout** approach. Klíčem je důkladné UAC testing a refactoring protected file system paths. S AI asistencí lze migraci dokončit za **€21,840 (5.5 týdnů)** místo €31,200 (8 týdnů).*

---

**Vygenerováno:** 2025-12-01 **Analýzoval:** Claude Code AI + Neo4J + Qdrant **Senior Full-Stack Java Developer Perspective**