

# Einsatzgebiete und Fehlerbehandlung

## Einsatzgebiete

ISSP verwendet die PowerShell Cmdlets generell meist in Verbindung mit Migrationen von Systemen, Transformation von Daten, Konvertierung etc. Einsatzgebiete sind hier kleinere Volumina sowie Volumina bis 10-50 Millionen Dokumenten und Verarbeitung von langläufigen Prozessen. Die Verarbeitung der Dokumente findet in Verbindung mit den anderen Cmdlet-Bibliotheken statt - etwa wie der Sharepoint/Migrations Cmdlet Bibliothek. Hier können dann Sharepoint Dokumente übernommen werden, oder aber bereits migrierte Dokumente in einer Migrationsdatenbank abgelegt werden, die auf dem ORM „XPO“ von DevExpress beruht.

Des Weiteren verwendet ISSP die Cmdlets für sogenanntes „Fast-Laning“ in der Entwicklung. Sprich wenn noch keine andere Routine den benötigten Funktionsumfang bietet, dann wird mittels PowerShell die Lücke hin zum Standard geschlossen. Das ermöglicht den schnelleren Durchsatz in der Kundenentwicklung und vermeidet höhere Aufwände in der Standardentwicklung.

## Speicherlecks von Cmdlets/PowerShell ISE

Sowohl die Cmdlets als auch die PowerShell ISE gelten als sehr stabil und weisen in der Regel keine Weariness/Fatigue (Performance Degradation) Effekte, sowie Speicherlecks oder plötzliche Fehler auf. Besondere Vorsicht sollte man aber dann walten lassen, wenn man direkt mit Speicheroperationen (Streams etc.) operiert oder Operationen an Bilddateien mit GDI+ durchführt.

## 500er Fehler durch den ELO IX

Diese Fehler können immer wieder vorkommen und zwar in folgenden Zusammenhang:

- leere Dokumentdateien: wird an Add-ELODoc ein leerer Stream/Datei übergeben gibt es einen 500er Fehler. Hier ist es eine gute Variante die Dateien auf ihre Länge zu überprüfen.
- zu große Dateien: steht dem IX ein Proxy vor, dann kann es bei zu großen Dateien ebenfalls einen 500er Fehler geben. Hier ist dann die maximale Requestgröße am Proxy (IIS, Apache etc.) dem entsprechend anzupassen
- fehlender Plattenplatz: auch das kommt bei Migrationen immer wieder einmal vor. Nach einer bestimmten Anzahl von migrierten Dokumenten, läuft die Festplatte voll.
- fehlerhafte Dokumentenpfade: teilweise kommt es bei fehlerhaften Konfigurationen vor, dass es Dok.Pfade gibt, die nicht mehr gültig sind. Ablagen dahin führen ebenfalls ggf. zu einem 500er Fehler.
- Session-Probleme: bei Cluster-Konfigurationen oder fehlerhaften Konfigurationen kann es sein, dass es hier zu fehlerhaften Umleitungen auf andere IndexServer kommt. Das führt dazu, dass andere IndexServer die Anfragen ablehnen, weil sie die Session nicht kennen.

Leider ist es meist auf Clientseite nicht möglich die Details des Fehlers zu bekommen, da die Apache Implementation des Tomcat mittels Hardening abgesichert ist.

Somit ist es hier immer erforderlich das IndexServer Log / Proxy Log zu analysieren. Bei mehreren IndexServern ggf. auch die Logs aller IndexServer.

Eine Alternative zu einer händischen Kontrolle ist eine automatisierte Kontrolle des Logs mittels zusätzlicher Tools oder eine Konfiguration der Logback.xml Datei für ein separates Fehlerlog.

- Beispiel eines Tails:  
<https://www.middlewareinventory.com/blog/powershell-tail-file-windows-tail-command/>
- Alternativ kann hier auch das „TailNet“ Tool des DDI/FTP4ELO verwendet werden (Parameter Dateiname+ERROR+(500))
- Auch eine Variante ist die Installation von Linux Tools wie „tail -f“ + „grep“

From:  
<https://powershell.issp.gmbh/> - **DokuWiki - ISSP PowerShell Module**

Permanent link:  
[https://powershell.issp.gmbh/doku.php?id=swxps4elo:usage\\_and\\_error\\_handling](https://powershell.issp.gmbh/doku.php?id=swxps4elo:usage_and_error_handling)

Last update: **2025/12/03 14:19**

