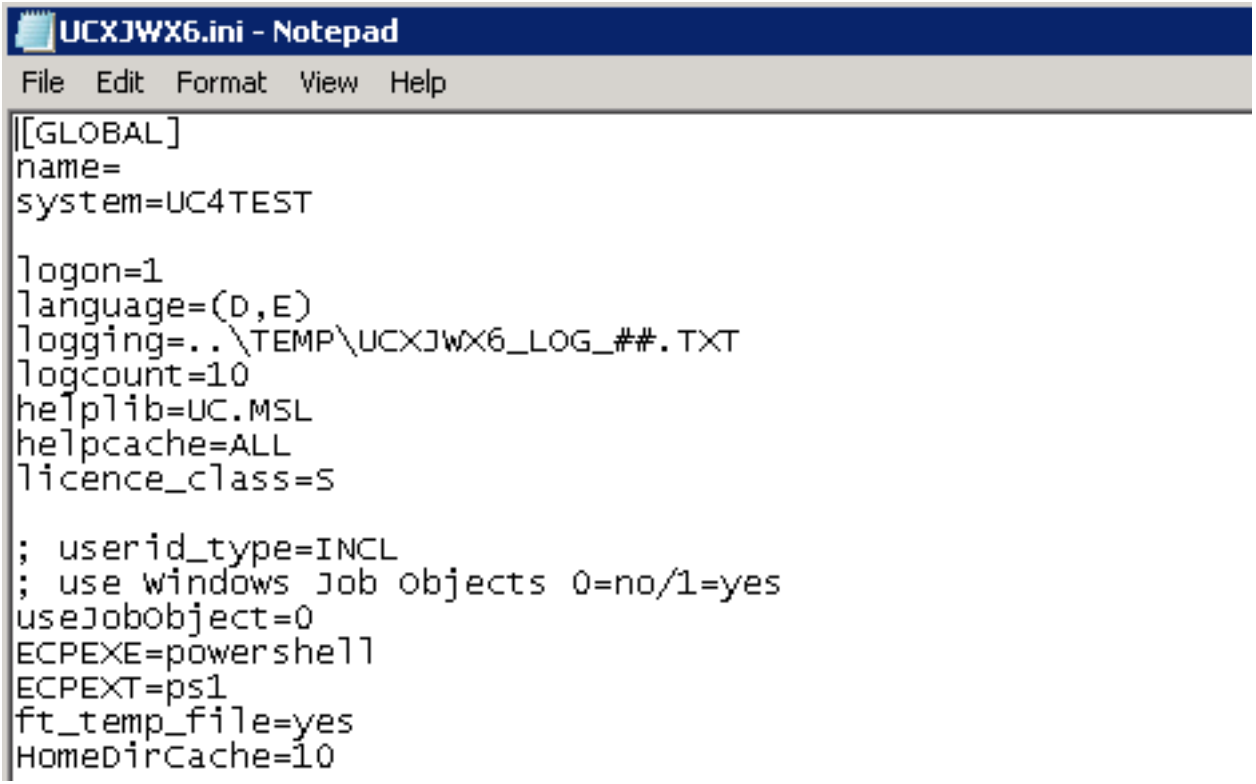
1.Aktivierung von Powershell als zusätzlichen Kommandointerpreter im Windows-Agenten

Evtl. den Agentennamen zusätzlich um den Text „_PS“ oder „_POWERSHELL“ erweitern, dann kann man die Änderungen in den später anzupassenden UC4-Includes besser greifen.



2.Berechtigung zur Ausführung von Powershell-Skripten für den Agenten vergeben

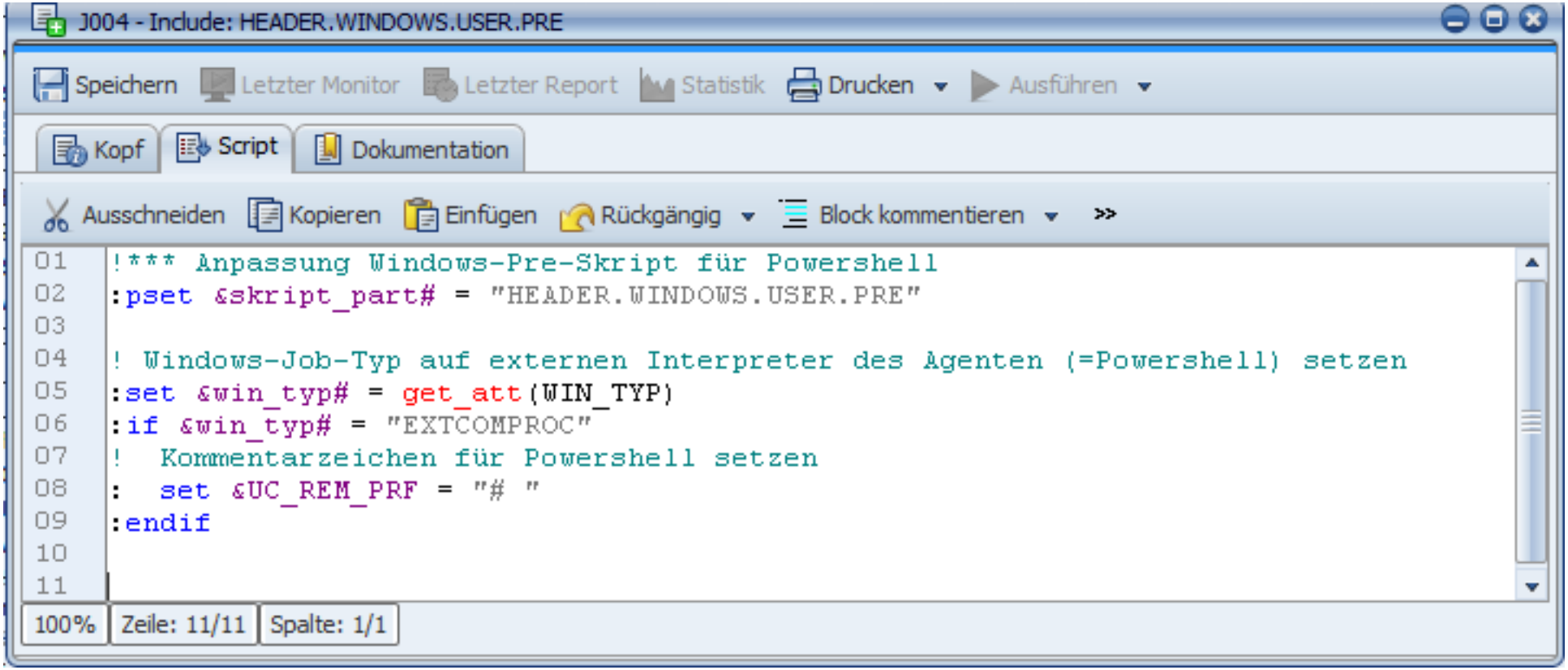
Standardmässig dürfen von dem UC4-Windowsagenten keine Powershellskripten ausgeführt werden.

Mit folgendem Befehl in der mit Adminrechten gestarteten Powershell vergibt man die Ausführungsrechte für lokale Skripte.

```
Set-ExecutionPolicy -Scope LocalMachine -ExecutionPolicy RemoteSigned
```

3. Anpassungen für die Skriptverarbeitung in Windows-Jobs

3.1. Festlegung des Kommentarzeichens für Powershell-Skripte



3.2. Festlegung einiger Ausgabeparameter / Fehlerhandling von Powershellskripten

Die Ausgabe der Powershell wird für die übliche Grösse eines Terminalfensters formatiert.
Ab Spalte 80 wird umgebrochen und alle 25 Zeilen erfolgt ein Seitenumbruch.

Um die Anzahl der Umbrüche zu reduzieren wird hier mit Host.Size die Grösse auf 2500 Zeilen und 2500 Spalten

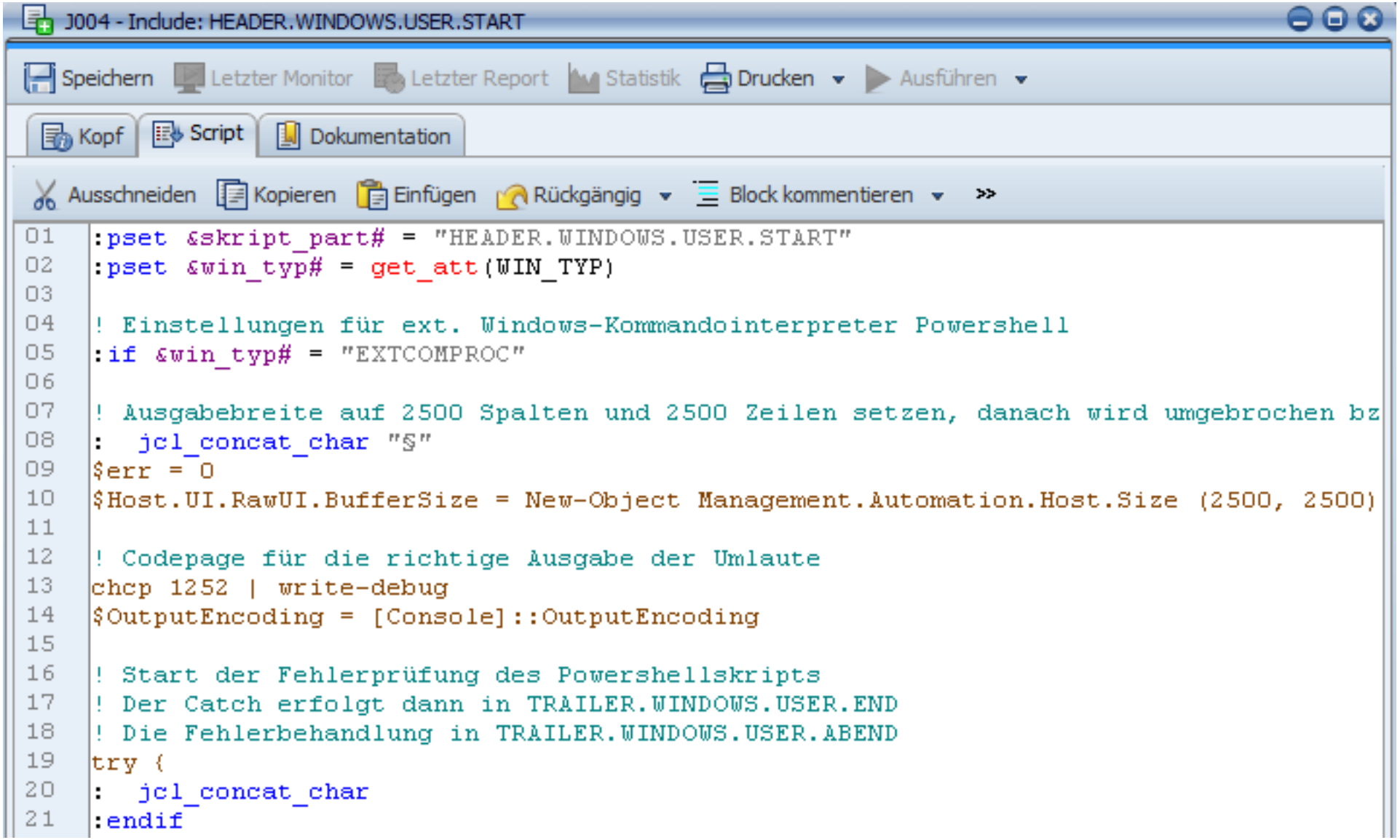
geändert.

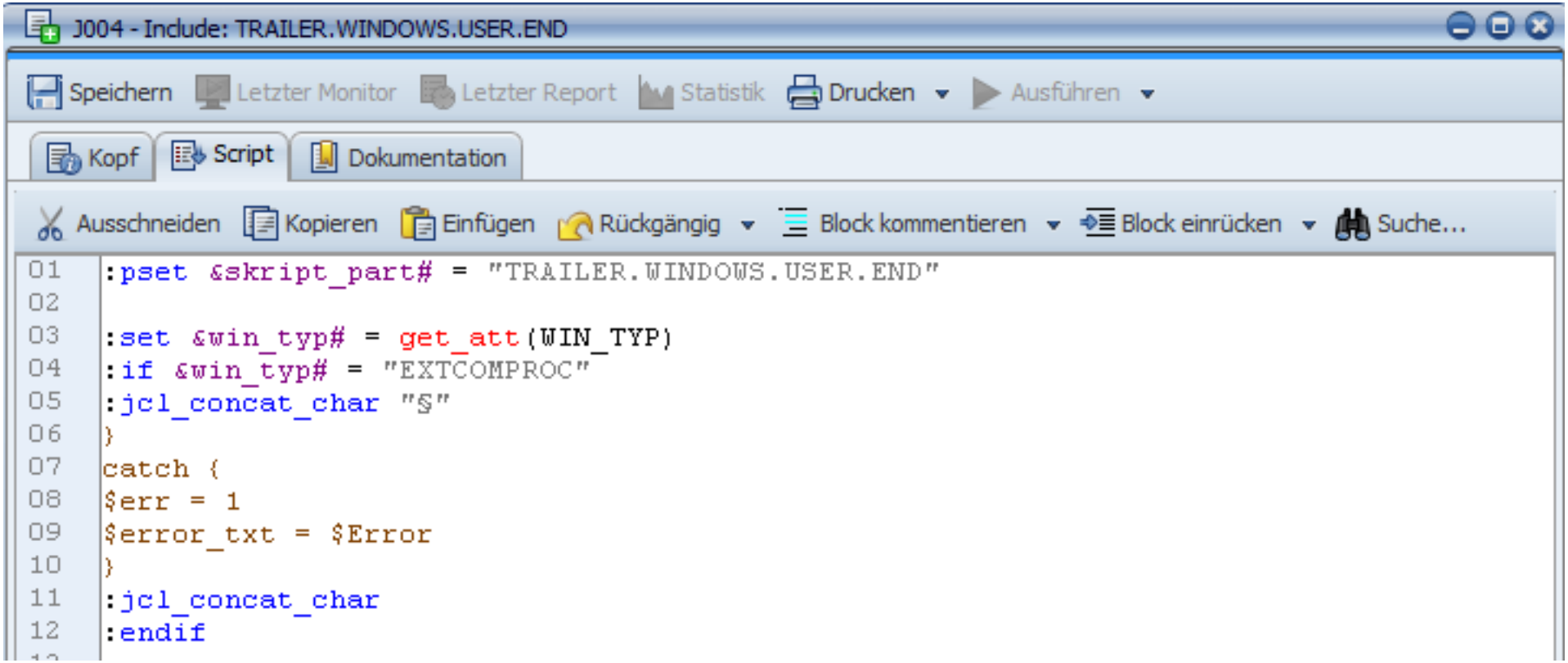
Mit der Darstellung von Umlauten gibt es üblicherweise Probleme. Bei Standeinstellung werden bei Ausgaben in die Konsole die Umlaute nicht ausgegeben.
Durch das DOS-Kommando „chcp 1252“ wird die Codepage für die Console geändert und durch anschliessende Zuweisung
an die (magische) Variable $OutputEncoding an Powershell übergeben.
Damit die Ausgabe des CHCP-Befehls nicht im Report erscheint, wir ein „| write-debug“ angehängt.

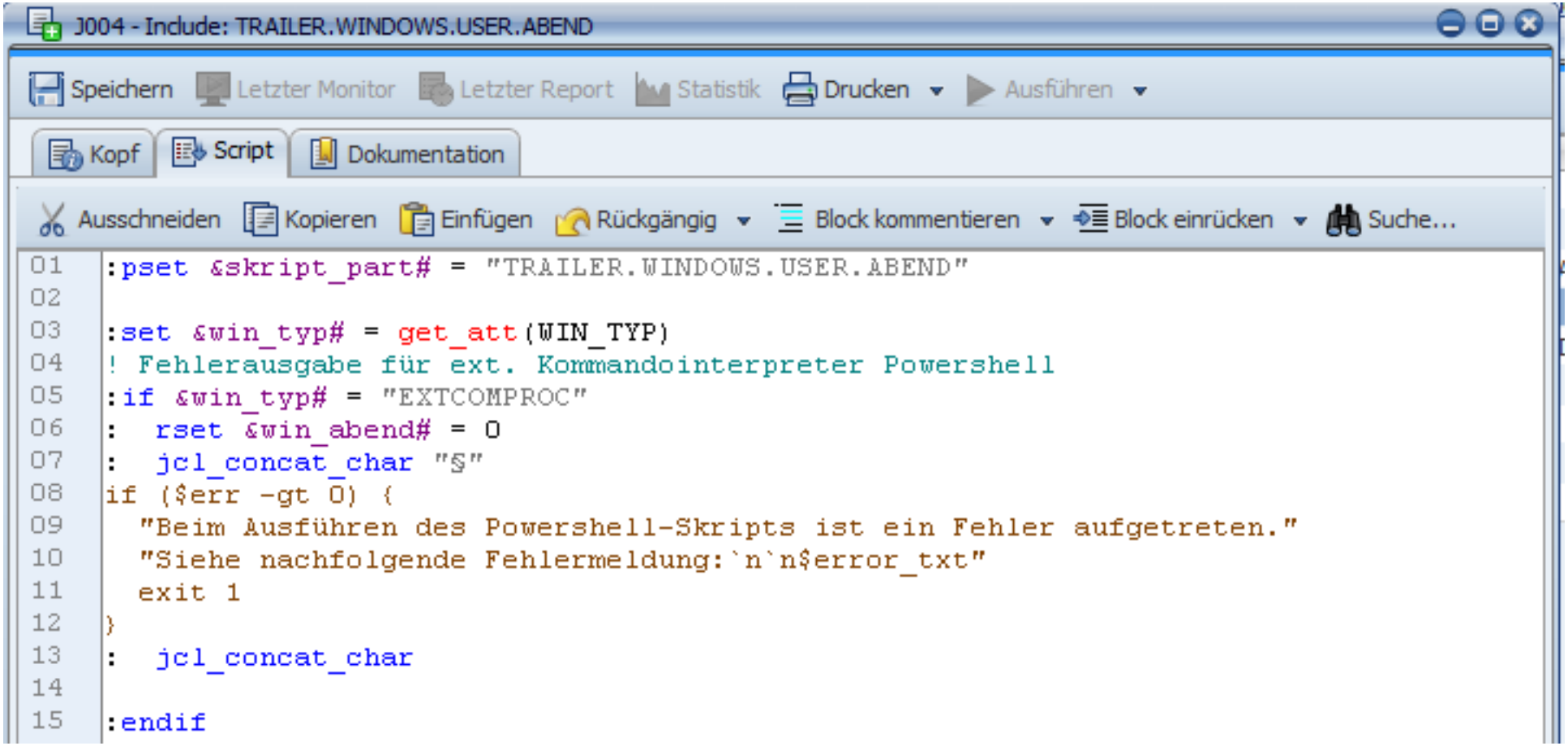
Die Fehlerbehandlung in UC4 berücksichtigt nur BATCH-Fehler. Für Fehler in Powershell wird mit
HEADER.WINDOWS.USER.START
und TRAILER.WINDOWS.USER.END ein Try/Catch-Block um das im Reiter ‚Skript‘ erstellte Powershellskript gebaut.

In TRAILER.WINDOWS.USER.ABEND erfolgt dann die Reaktion auf den Fehler. Dazu wird im Fehlerfall im Catch-Block von
TRAILER.WINDOWS.USER.END die Powershell-Variable $err auf 1 gesetzt und in $err_text der Fehlertext gespeichert.

Der Abbruch im Fehlerfall erfolgt in TRAILER.WINDOWS.USER.ABEND mit dem DOS-Befehl ‚EXIT 1‘.
Zuvor wird noch die in $err_text gespeicherte Fehlermeldung im Report ausgegeben.

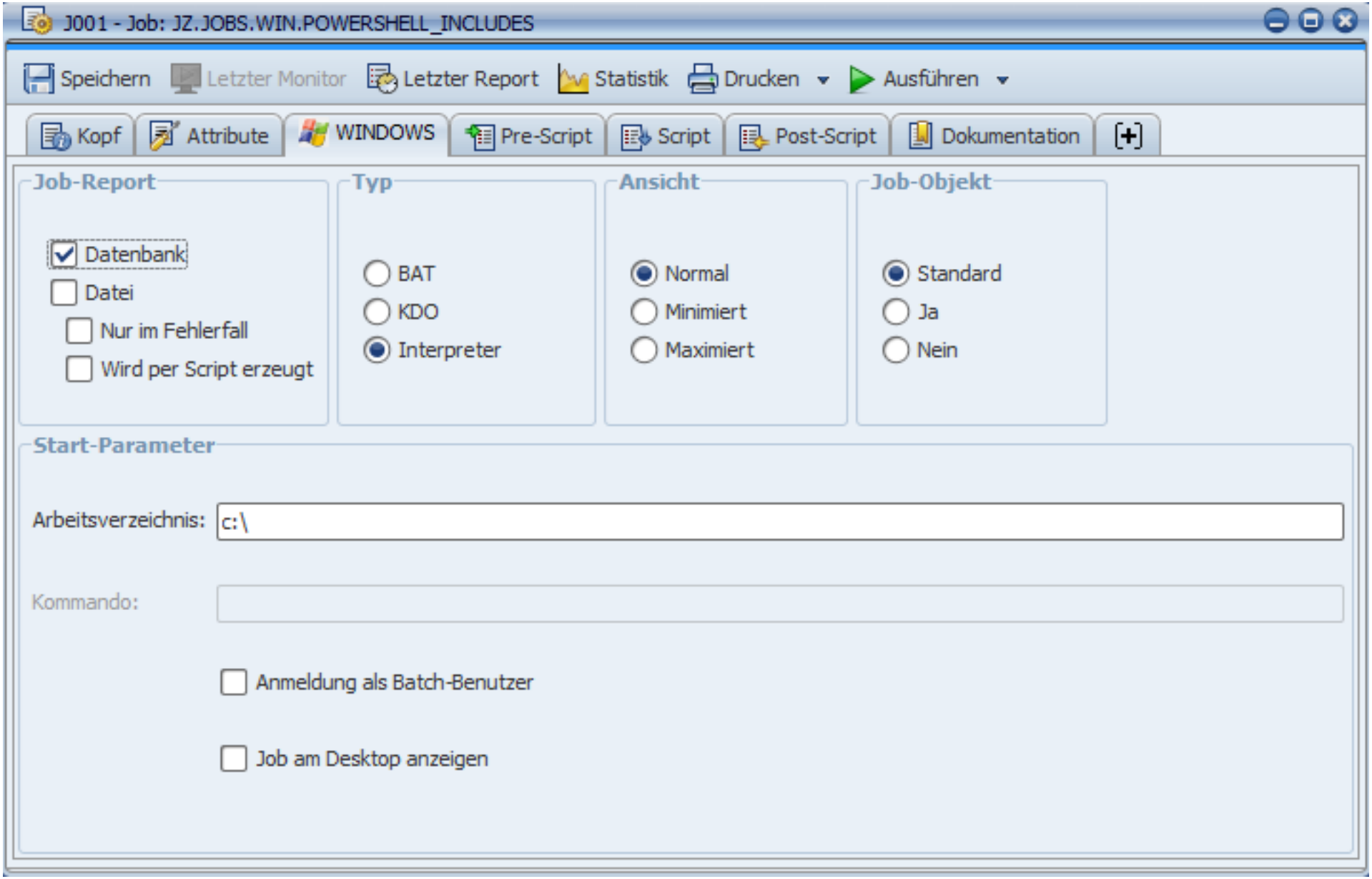

J004 - Include: HEADER.WINDOWS.USER.START
Speichern
Letzter Monitor
Letzter Report
Statistik
Drucken
Ausführen
Kopf
Script
Dokumentation
Ausschneiden
Kopieren
Einfügen
Rückgängig
Block kommentieren
>>
01 :pset &skript_part# = "HEADER.WINDOWS.USER.START"
02 :pset &win_typ# = get_att(WIN_TYP)
03
04 ! Einstellungen für ext. Windows-Kommandointerpreter Powershell
05 :if &win_typ# = "EXTCOMPROC"
06
07 ! Ausgabebreite auf 2500 Spalten und 2500 Zeilen setzen, danach wird umgebrochen bz
08 : jcl_concat_char "§"
09 $err = 0
10 $Host.UI.RawUI.BufferSize = New-Object Management.Automation.Host.Size (2500, 2500)
11
12 ! Codepage für die richtige Ausgabe der Umlaute
13 chcp 1252 | write-debug
14 $OutputEncoding = [Console]::OutputEncoding
15
16 ! Start der Fehlerprüfung des Powershellskripts
17 ! Der Catch erfolgt dann in TRAILER.WINDOWS.USER.END
18 ! Die Fehlerbehandlung in TRAILER.WINDOWS.USER.ABEND
19 try {
20 : jcl_concat_char
21 :endif


J004 - Include: TRAILER.WINDOWS.USER.END
Speichern
Letzter Monitor
Letzter Report
Statistik
Drucken
Ausführen
Kopf
Script
Dokumentation
Ausschneiden
Kopieren
Einfügen
Rückgängig
Block kommentieren
Block einrücken
Suche...
01 :pset &skript_part# = "TRAILER.WINDOWS.USER.END"
02
03 :set &win_typ# = get_att(WIN_TYP)
04 :if &win_typ# = "EXTCOMPROC"
05 :jcl_concat_char "§"
06 }
07 catch {
08 $err = 1
09 $error_txt = $Error
10 }
11 :jcl_concat_char
12 :endif



4. Verwendung der Powershell in einem Windows-Job

Im Reiter ‚Windows‘ muss der ext.Kommandointerpreter als Typ Interpreter aktiviert werden.


J001 - Job: JZ.JOBS.WIN.POWERSHELL_INCLUDES
Speichern
Letzter Monitor
Letzter Report
Statistik
Drucken
Ausführen
Kopf
Attribute
WINDOWS
Pre-Script
Script
Post-Script
Dokumentation
Job-Report
Datenbank
Datei
Nur im Fehlerfall
Wird per Script erzeugt
Typ
BAT
KDO
Interpreter
Ansicht
Normal
Minimiert
Maximiert
Job-Objekt
Standard
Ja
Nein
Start-Parameter
Arbeitsverzeichnis:
c:\
Kommando:
Anmeldung als Batch-Benutzer
Job am Desktop anzeigen

Dann kann im Reiter ‚Script' Powershell-Code verwendet werden.
Hier wird z.B. statt get-help das nicht vorhandene Commandlet get-helpx aufgerufen.
Der Aufruf führt zu einem Jobabbruch.