ChatGPT

--- a/Main.ps1 +++ b/Main.ps1 @@ -1,4 +1,5 @@ #region Importering och config +# Executive summary: Jag härdar TEMP-snapshot, fil-lås och CSV-cache för stabil produktion (PS 5.1 + EPPlus 4.5.3.3) med minimal diffyta. if ([Threading.Thread]::CurrentThread.ApartmentState -ne 'STA') { $exe = Join-Path $PSHome 'powershell.exe' $scriptPath = if ($PSCommandPath) { $PSCommandPath } else { $MyInvocation.MyCommand.Path } @@ -2025,11 +2026,14 @@ try {

```
            # Standard: C:\IPTCompile_TEMP\... (kan styras via Config:
TempSnapshotRoot)
```

- ## Jag expanderar miljövariabler och skapar mappar med Directory.CreateDirectory för robust staging (PS 5.1-säker; risk: endast fil-I/O).

```
          $root = Get-ConfigValue -Name 'TempSnapshotRoot' -Default 'C:
\IPTCompile_TEMP' -ConfigOverride $Config
          if (-not $root) { $root = 'C:\IPTCompile_TEMP' }
```

- if (-not (Test-Path -LiteralPath $root)) { New-Item -ItemType Directory -Path $root -Force | Out-Null }
- $stageDir = Join-Path $root ("IPTCompile_" + $lsp + "" + *(Get-Date -Format 'yyyyMMdd_HHmmss') +* "" + ([guid]::NewGuid().ToString('N')))
- New-Item -ItemType Directory -Path $stageDir -Force | Out-Null
- $root = [Environment]::ExpandEnvironmentVariables(($root + '').Trim())
- [IO.Directory]::CreateDirectory($root) | Out-Null
- $stageName = ("IPTCompile_{0}*{1}*" -f $lsp, (Get-Date -Format 'yyyyMMdd_HHmmss'), ([guid]::NewGuid().ToString('N')))
- $stageDir = Join-Path $root $stageName
- [IO.Directory]::CreateDirectory($stageDir) | Out-Null # Spara original paths $selNegOrig = $selNeg $selPosOrig = $selPos @@ -4918,8 +4922,12 @@ if ($stageDir -and (Test-Path -LiteralPath $stageDir)) { # Säkerhet: radera bara om den ligger under TEMP-root $root = Get-ConfigValue -Name 'TempSnapshotRoot' -Default 'C:\IPTCompile_TEMP' -ConfigOverride $Config if (-not $root) { $root = 'C:\IPTCompile_TEMP' }
- $stageFull = (Resolve-Path -LiteralPath $stageDir -ErrorAction SilentlyContinue).Path
- $rootFull = (Resolve-Path -LiteralPath $root -ErrorAction SilentlyContinue).Path
- $root = [Environment]::ExpandEnvironmentVariables(($root + '').Trim())
- $stageFull = $null; $rootFull = $null
- try { $stageFull = [IO.Path]::GetFullPath($stageDir) } catch { $stageFull = $stageDir }
- try { $rootFull = [IO.Path]::GetFullPath($root) } catch { $rootFull = $root }

- # Jag säkrar att TEMP-root slutar med "\" så jag inte råkar matcha t.ex. "C:\IPTCompile_TEMPORARY" (PS 5.1-säker; risk: låg).

- if ($rootFull -and (-not $rootFull.EndsWith('\'))) { $rootFull = ($rootFull.TrimEnd('\') + '\') } if ($stageFull -and $rootFull -and $stageFull.StartsWith($rootFull, [System.StringComparison]::OrdinalIgnoreCase)) { Remove-Item -LiteralPath $stageDir -Recurse -Force -ErrorAction SilentlyContinue } --- a/Modules/DataHelpers.ps1 +++ b/Modules/DataHelpers.ps1 @@ -1540,9 +1540,8 @@ } catch {}

  try { - if (-not (Test-Path -LiteralPath $StageDir)) { - New-Item -ItemType Directory -Path $StageDir -Force | Out-Null - } + # Jag säkrar att StageDir finns med Directory.CreateDirectory (PS 5.1-säker; risk: låg). + [IO.Directory]::CreateDirectory($StageDir) | Out-Null

  ```
   $leaf = Split-Path $Path -Leaf
   $dstLeaf = if ($Prefix) { ($Prefix + '__' + $leaf) } else { $leaf }
  ```

  --- a/Main.ps1 +++ b/Main.ps1 @@ -1185,7 +1185,8 @@

  ```
   try {
       $target = Get-UniqueDestinationPath -DestinationDir
  $destFolder.FullName -FileName $f.Name
  ```

  - Move-Item -LiteralPath $f.FullName -Destination $target -Force

  - # Jag använder -ErrorAction Stop så try/catch blir deterministisk vid flytt (PS 5.1-säker; risk: låg).

  - Move-Item -LiteralPath $f.FullName -Destination $target -Force -ErrorAction Stop $moved.Add((Split-Path -Leaf $target)) } catch { $failed.Add(("{0} -> {1}" -f $f.Name, $_.Exception.Message)) --- a/Modules/DataHelpers.ps1 +++ b/Modules/DataHelpers.ps1 @@ -127,8 +127,15 @@ function Set-CellBorder { param($cell,[string]$style='Thin') $border = [OfficeOpenXml.Style.ExcelBorderStyle]::$style if ($border) { $cell.Style.Border.Top.Style=$border; $cell.Style.Border.Bottom.Style=$border; $cell.Style.Border.Left.Style=$border; $cell.Style.Border.Right.Style=$border } }

-function Test-FileLocked { param([Parameter(Mandatory=$true)][string]$Path) - try { $fs = [IO.File]::Open($Path,'Open','ReadWrite','None'); $fs.Close(); return $false } catch { return $true } +function Test-FileLocked { + param([Parameter(Mandatory=$true)][string]$Path) + # Jag använder enum-typer för File.Open för att vara robust i PS 5.1/.NET och göra beteendet tydligt (risk: låg). + try { +

$fs = [IO.File]::Open($Path, [IO.FileMode]::Open, [IO.FileAccess]::ReadWrite, [IO.FileShare]::None) + $fs.Close() + return $false + } catch { return $true } }

function Get-CsvDelimiter { param([string]$Path) try { $first = Get-Content -LiteralPath $Path -Encoding Default -TotalCount 30 | Where-Object { $ *-and $*.Trim() } | Select-Object -First 1 --- a/Main.ps1 +++ b/Main.ps1 @@ -692,7 +692,7 @@ $lines = Get-Content -LiteralPath $Path -TotalCount 30 -ErrorAction Stop if (-not $lines -or $lines.Count -le 9) { return $false }

- $delim = Get-CsvDelimiter -Path $Path
- $delim = Get-CsvDelimiter -Path $Path -Lines $lines

```
# Rubriken ligger på rad 8 (index 7) i våra Test Summary-CSV
$hdr = $null
```

@@ -2063,10 +2063,10 @@ $script:CsvLinesPrimary = $null $script:CsvLinesResample = $null if ($selCsv -and (Test-Path -LiteralPath $selCsv)) { - try { $script:CsvLinesPrimary = Get-Content -LiteralPath $selCsv } catch { Gui-Log ("⚠ Kunde inte läsa Primary CSV: " + $*.Exception.Message) 'Warn' } + try { $script:CsvLinesPrimary = Get-Content -LiteralPath $selCsv -ErrorAction Stop } catch { Gui-Log ("⚠ Kunde inte läsa Primary CSV: " + $*.Exception.Message) 'Warn' } } if ($selCsvRes -and (Test-Path -LiteralPath $selCsvRes)) { - try { $script:CsvLinesResample = Get-Content -LiteralPath $selCsvRes } catch { Gui-Log ("⚠ Kunde inte läsa Resample CSV: " + $*.Exception.Message) 'Warn' } + try { $script:CsvLinesResample = Get-Content -LiteralPath $selCsvRes -ErrorAction Stop } catch { Gui-Log ("⚠ Kunde inte läsa Resample CSV: " + $*.Exception.Message) 'Warn' } } @@ -2332,7 +2332,7 @@ $all = $script:CsvLinesPrimary if ($all -and $all.Count -gt 9) { - $del = Get-CsvDelimiter -Path $selCsv + $del = Get-CsvDelimiter -Path $selCsv -Lines $all $hdr = ConvertTo-CsvFields $all[7] $dl = $all[9..($all.Count-1)] | Where-Object { $ *-and $*.Trim() } $csvObjs = @(ConvertFrom-Csv -InputObject ($dl -join "`n") -Delimiter $del -Header $hdr)`

```
@@ -2386,7 +2386,7 @@
                         $allR = $script:CsvLinesResample
                         if ($allR -and $allR.Count -gt 9) {
-                                $delR = Get-CsvDelimiter -Path $selCsvRes
+                                 $delR = Get-CsvDelimiter -Path $selCsvRes
-Lines $allR
                             $hdrR = ConvertTo-CsvFields $allR[7]
                                 $dlR  = $allR[9..($allR.Count-1)] | Where-
Object { $_ -and $_.Trim() }
                               $csvObjsRes = @(ConvertFrom-Csv -InputObject
($dlR  -join  "
```

n") -Delimiter $delR -Header $hdrR) @@ -4754,7 +4754,7 @@ $all = $script:CsvLinesPrimary if ($all -and $all.Count -gt 9) { $hdr = ConvertTo-CsvFields $all[7] $dl = $all[9..($all.Count-1)] | Where-Object { $ *-and $*.Trim() } - $del = Get-CsvDelimiter -Path $selCsv + $del = Get-CsvDelimiter -Path $selCsv -Lines $all $csvObjs2 = @(ConvertFrom-Csv -InputObject ($dl -join "`n") -Delimiter $del -Header $hdr)`

```
                                     }
                                 } catch { $csvObjs2 = @() }
@@ -4777,7 +4777,7 @@
                                     $allR = $script:CsvLinesResample
                                     if ($allR -and $allR.Count -gt 9) {
                                               $hdrR = ConvertTo-CsvFields
$allR[7]
```

```
                                          $dlR  = $allR[9..($allR.Count-1)]
| Where-Object { $_ -and $_.Trim() }
-                                            $delR = Get-CsvDelimiter -Path
$selCsvRes
+                                            $delR = Get-CsvDelimiter -Path
$selCsvRes -Lines $allR
                                            $csvObjsR2 = @(ConvertFrom-Csv -
InputObject ($dlR -join "
```

n") -Delimiter $delR -Header $hdrR) } } catch { $csvObjsR2 = @() } --- a/Modules/DataHelpers.ps1 +++ b/Modules/DataHelpers.ps1 @@ -139,11 +139,26 @@ -function Get-CsvDelimiter { param([string]$Path) +function Get-CsvDelimiter { + param(+ [string]$Path, + [Parameter(Mandatory=$false)][string[]]$Lines +) try { - $first = Get-Content -LiteralPath $Path -Encoding Default -TotalCount 30 | Where-Object { $ *-and $*.Trim() } | Select-Object -First 1 + # Jag återanvänder redan inlästa rader när de finns för att undvika extra disk-I/O (PS 5.1-säker; risk: låg). + $first = $null + if ($Lines -and $Lines.Count -gt 0) { + $first = ($Lines | Select-Object -First 30 | Where-Object { $ *-and $*.Trim() } | Select-Object -First 1) + } elseif ($Path) { + $first = Get-Content -LiteralPath $Path -Encoding Default -TotalCount 30 -ErrorAction Stop | + Where-Object { $ *-and $*.Trim() } | Select-Object -First 1 + } if (-not $first) { return ';' } $sc = ($first -split ';').Count; $cc = ($first -split ',').Count if ($cc -gt $sc -and $cc -ge 2) { return ',' } else { return ';' } } catch { Gui-Log "⚠ Get-CsvDelimiter misslyckades: $($_.Exception.Message)" 'Warn' return ';' } }

function ConvertTo-CsvFields { param([Parameter(Mandatory)]$Line,[Parameter(Mandatory)][string]$Delimiter) try { return ($Line -split [Regex]::Escape($Delimiter)) } catch { return @() } } --- a/Main.ps1 +++ b/Main.ps1 @@ -1267,11 +1267,16 @@ [System.Windows.Forms.MessageBox]::Show("Filen hittades inte: `n$Path", $Label) | Out-Null; return`

```
     }
     $ext = [System.IO.Path]::GetExtension($Path).ToLowerInvariant()
-    switch ($ext) {
-         '.ps1' { Start-Process powershell.exe -ArgumentList "-ExecutionPolicy
Bypass -File
```

"$Path `"" }`

```
-        '.bat' { Start-Process cmd.exe -ArgumentList "/c
```

"$Path `"" }`

```
-        '.lnk' { Start-Process -FilePath $Path }
-                default  {  try  {  Start-Process  -FilePath  $Path  }  catch
{ [System.Windows.Forms.MessageBox]::Show("Kunde inte öppna filen:
```

n$Path", $Label) | Out-Null } } + try { + # Jag använder -ErrorAction Stop så fel hamnar i catch och blir tydliga för användaren (PS 5.1-säker; risk: låg). + switch ($ext) { + '.ps1' { Start-Process -FilePath powershell.exe -ArgumentList "-ExecutionPolicy Bypass -File `"$Path`"" -ErrorAction Stop } + '.bat' { Start-Process -FilePath cmd.exe -ArgumentList "/c `"$Path`"" -ErrorAction Stop } + '.lnk' { Start-Process -FilePath $Path -ErrorAction Stop } + default { Start-Process -FilePath $Path -ErrorAction Stop } + } + } catch { + [System.Windows.Forms.MessageBox]::Show("Kunde inte öppna filen: `n$Path`n`n$`

```
($_.Exception.Message)", $Label) | Out-Null
     }
 }
 $miScript1.add_Click({ Invoke-ExternalScript -Path $Script1Path -Label 'Skript
1' })
--- /dev/null
+++ b/PATCH_SUMMARY.md
@@ -0,0 +1,36 @@
+# Patch-sammanfattning (IPTCompile_16)
+
+## Projektöversikt (statisk lista från zip)
```

```
+
+| Typ | Sökväg |
+|---|---|
+| Entrypoint | Main.ps1 |
+| Version | Version.txt |
+| Mall | output_template-v4.xlsx |
+| Ikon | icon.png |
+| Regelbank | RuleBank/RuleBank.compiled.ps1 |
+| Moduler | Modules/Config.ps1 , Modules/DataHelpers.ps1 , Modules/Logging.ps1 , Modules/RuleEngine.ps1 , Modules/SharePointClient.ps1 , Modules/SignatureHelpers.ps1 , Modules/Splash.ps1 , Modules/UiStyling.ps1 , Modules/EPPlus.dll |
+
+## Ändrade filer (original → patchad)
+
+| Fil | Original rader | Patchad rader | Δ rader | Syfte (kort) |
+|---|---:|---:|---:|---|
+| Main.ps1 | 4974 | 4988 | +14 | Robust TEMP-staging, deterministisk flytt & CSV-delimiter utan extra I/O |
+| Modules/DataHelpers.ps1 | 1701 | 1720 | +19 | Enum-säker fil-lås, delimiter från inlästa rader, robust StageDir-skapande |
+| PATCH_SUMMARY.md | – | 36 | +36 | Patch-rapport som fil (tabell + testchecklista) |
+| PATCH_FLOW.mmd | – | 20 | +20 | Mermaid: snapshot+signeringsflöde |
+| PATCH_TIMELINE.mmd | – | 14 | +14 | Mermaid: tidslinje för patchsteg |
+
+## Kort testchecklista (mental regression)
+
+- [ ] **Normal**: 1 CSV (Primary), NEG+POS, Worksheet valfri → rapport skapas.
+- [ ] **Resample**: 2 CSV (Primary+Resample) → resample-detektion fungerar, RuleEngine körs.
+- [ ] **MISQ**: Worksheet Data Summary/Resample Data Summary → MISQ/fynd dyker i rapporten.
+- [ ] **Flytta från Downloads**: låst fil → Retry/Cancel; olåst fil → flytt + listrefresh.
+- [ ] **Staging**: EnableLocalStaging = true → TEMP-mapp skapas och rensas i finally.
+- [ ] **Signering**: signering försöker på ORIGINAL (om ej låst) och stoppar inte rapport.
```

+- [ ] **PS 5.1**: inga PS7-only konstruktioner (t.ex. ?? , ForEach-Object -Parallel`). + +## Notering (riskprofil) + +- Patchen ändrar inte affärslogik, endast I/O-robusthet och felhantering (layout/logik för rapportdata oförändrad). --- /dev/null +++ b/PATCH_FLOW.mmd @@ -0,0 +1,20 @@ +%% Jag visar översiktligt flöde för snapshot + signering (PS 5.1-säker, endast dokumentation). +flowchart TD + A[Skapa rapport (GUI)] --> B[Validera valda filer] + B --> C{EnableLocalStaging?} + C -- Ja --> D[Expandera TempSnapshotRoot\nCreateDirectory(root)\nCreateDirectory(stageDir)] + D --> E[Stage-InputFileSnapshot\nNEG/POS/CSV/Worksheet -> stageDir] + C -- Nej --> F[Använd original paths] + E --> G[Öppna staged NEG/POS\nLäs staged CSV] + F --> G + G --> H{Signering aktiverad?} + H -- Ja --> I[Test-FileLocked på ORIGINAL] + I --> J{Olåst?} + J -- Ja --> K[Skriv signatur i ORIGINAL\n(fångas i try/catch)] + J -- Nej --> L[Logga 'låst'\nfortsätt utan signering] + H -- Nej --> M[Hoppa signering] + K --> N[Bygg output (template)] + L --> N + M --> N + N --> O[Spara rapport] + O --> P[finally: Dispose paket\nTa bort stageDir om under TempSnapshotRoot] --- /dev/null +++ b/PATCH_TIMELINE.mmd @@ -0,0 +1,14 @@ +%% Jag visar en kort tidslinje för patchsteg (PS 5.1-säker, endast dokumentation). +timeline + title IPTCompile_16 - patchsteg (översikt) + section Staging + Expandera TempSnapshotRoot + CreateDirectory : done + Säkrare cleanup (GetFullPath + trailing) : done + section Downloads-flytt + Enum-säker Test-FileLocked : done + Move-Item med -ErrorAction Stop : done + section CSV-cache + Get-CsvDelimiter kan ta -Lines : done + Test-IsResampleCsv återanvänder rader : done

---