O plánované spuštění Excelu a sešitu se postará Plánovač úloh ve Windows, případně skript navázaný na událost. Pro spouštění maker můžeme jít dvojí cestou:

1) přímým voláním ze skriptu s pomocí známé metody Application.Run (ta podporuje i použití parametrů)
2a) prostým spuštěním sešitu, v němž je volání maker umístěno do událostní procedury Workbook_Open (historicky mírně odlišné makro Auto_Open)
2b) spuštěním sešitu s uvedením parametru /m, lépe řečeno /mNazevMakra (Command-line switches for Microsoft Office products)

Příklady níže aplikují první ze způsobů.

Jako skript se nabízí BAT soubory, VBScript či PowerShell. Zpracovány jsou druhé dva.

## VBScript

VBScript (viz také pojmy Windows Scripting Host, CScript) je tu už od nepaměti. Sloužil jako programovací jazyk pro Internet Explorer a je tu rovněž ve formě samospustitelných skriptů. Operační systém u souborů .vbs nezobrazoval příponu, a ty byly proto zneužívány pro nejrůznější havěť, hlavně v přílohách e-mailů. Přesto je tu tato podoba skriptů dostupná dodnes a hodí se pro naplánované úlohy a obsluhu sešitů Excelu (s makry).

Uvedený výpis postačí uložit (a spustit poklepáním) v prostém textovém editoru s příponou .vbs.

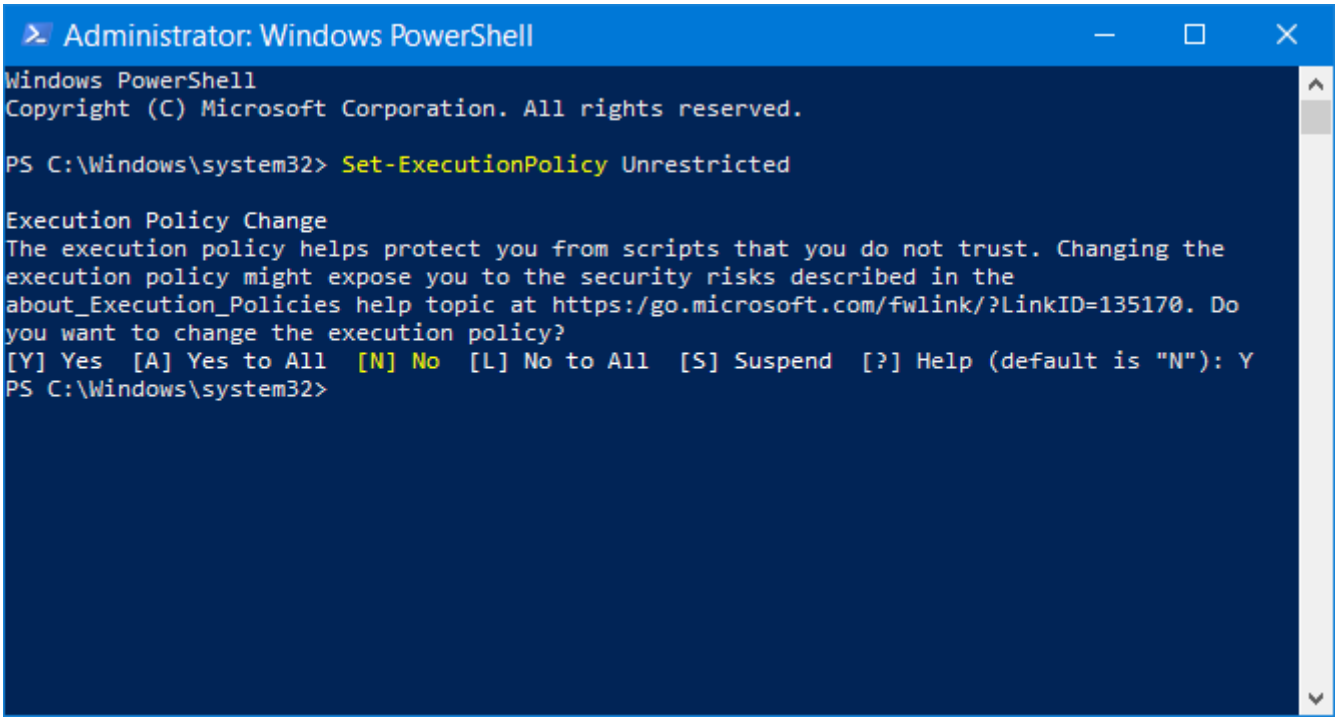
```
'VBScript

Set objShell = CreateObject("Wscript.Shell")
strCesta = objShell.CurrentDirectory

Set objExcel = CreateObject("Excel.Application")
Set objSesit = objExcel.Workbooks.Open(strCesta & "" & "sesit.xlsm")
Set objList = objSesit.Worksheets(1)

objExcel.DisplayAlerts = False

objExcel.Visible = True
objList.Cells(1, 1).Value = Now

objExcel.Run "MojeMakro"

objSesit.Save
objSesit.Close

objExcel.Quit
```

Kód je snadno čitelný i vzhledem k tomu, že VBScript a VBA mají společný základ z Visual Basicu 5 a 6. Vypnutí upozornění (DisplayAlerts) je nutné, jinak se běh zastaví na výzvě o uložení nového sešitu/listu s názvem List1. Ten vzniká pravděpodobně v rámci metody CreateObject, podobně jako když poklepete na zástupce aplikace Excel.

## PowerShell

Abychom ve Windows mohli spouštět nepodepsané skripty PowerShellu, musíme pro tuto činnost nastavit patřičná práva. To znamená kupříkladu spustit konzoli Windows PowerShell s právy správce a v ní následující kód:

```
Set-ExecutionPolicy Unrestricted
```




PowerShell – práva ke spouštění skriptů

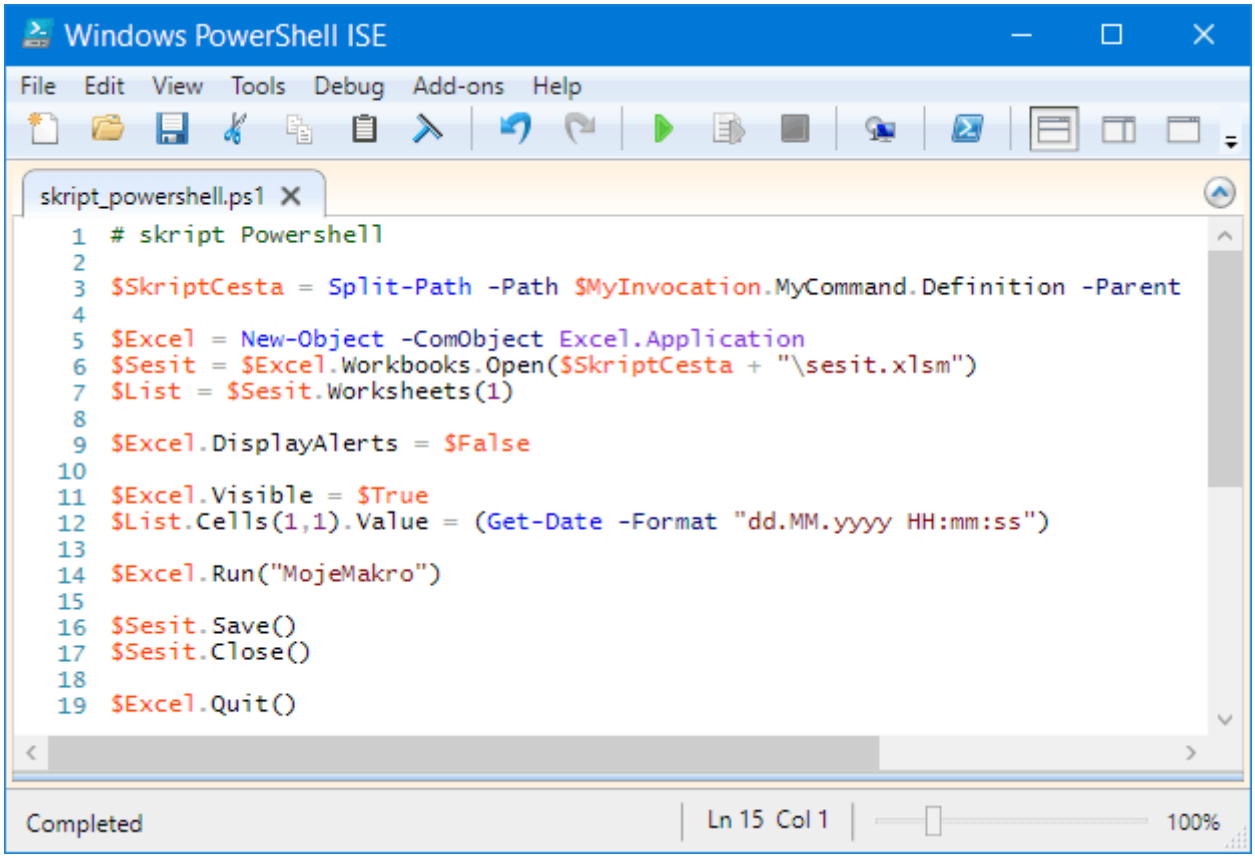
V zápisu níže se lze snadno orientovat, až tak výrazně se příkazy neliší od předchozích. Nicméně od mladšího nástroje, než je VBScript, bych čekal nějakou přidanou hodnotu. Ale nějak mě nadšení míjí. I když vyřešíme práva, pro mě nepřekonatelnou překážkou se stala taková hloupost, jako je zápis datumu do buňky Excelu. Ať jsem dělal, co dělal, datum se vždy vložil jako text. Takže jsem nakonec nechal původní syntaxi a nezabýval se dalšími pokusy (možná by pomohlo předformátování buňky a

zápisu data a času ve formě desetinného čísla). Při ukládání a zavírání sešitu mi chvíli trvalo, než jsem pochopil, že metody Save a Close vyžadují závorky (jinak se dočkáme tzv. OverloadDefinitions s hláškou typu void Save()). A stejně jako u VBScriptu, i zde platí, že v případě nevypnutí DisplayAlerts zůstane instance Excelu „viset“ a aplikace se neukončí.

Výpis níže postačí uložit jako obyčejný textový soubor s příponou .ps1. Nespouští se ovšem poklepáním, ale příkazem v konzoli PowerShellu.

```
# Powershell

$SkriptCesta = Split-Path -Path $MyInvocation.MyCommand.Definition -Parent

$Excel = New-Object -ComObject Excel.Application
$Sesit = $Excel.Workbooks.Open($SkriptCesta + "\sesit.xlsm")
$List = $Sesit.Worksheets(1)

$Excel.DisplayAlerts = $False

$Excel.Visible = $True
$List.Cells(1,1).Value = (Get-Date -Format "dd.MM.yyyy HH:mm:ss")

$Excel.Run("MojeMakro")

$Sesit.Save()
$Sesit.Close()

$Excel.Quit()
```

Pokud na uložený soubor .ps1 klepnete pravým tlačítkem myši, najdete v kontextové nápovědě volbu Upravit. Ta vás přesune do editoru Windows PowerShell ISE. Postupovat samozřejmě můžete i opačně. Nejprve spustíte editor a posléze vytváříte kód (viz View /Show Script Pane). Doporučuji použít také volbu Show Command Add-on.




PowerShell – editor a konzole

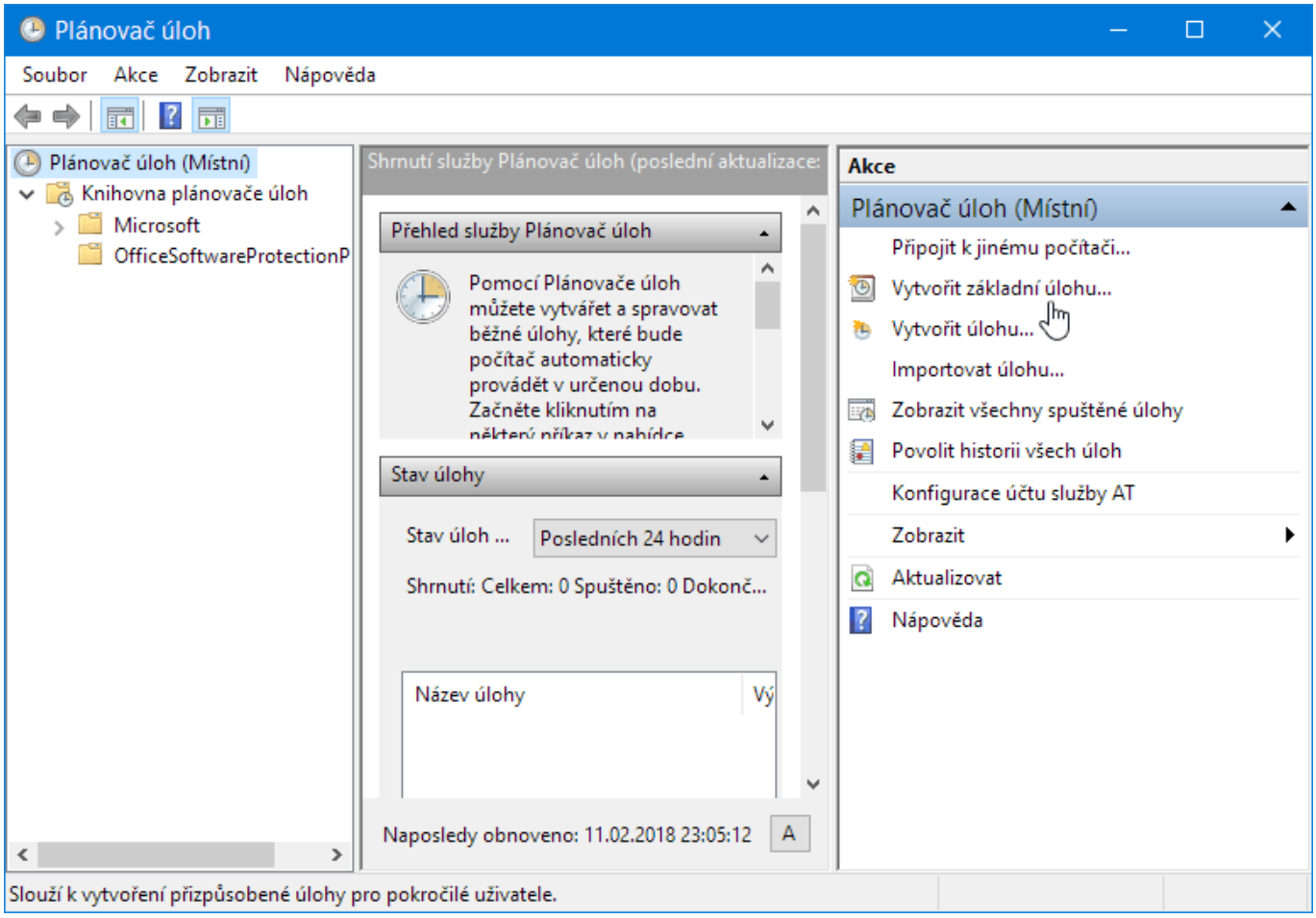
## Plánovač úloh

Ať už jeden či druhý typ skriptu máte vytvořen a zbývá naplánovat jejich spuštění ve Windows. Následuje obrazový průvodce pro PowerShell. Pro samotné spuštění je použit příkaz

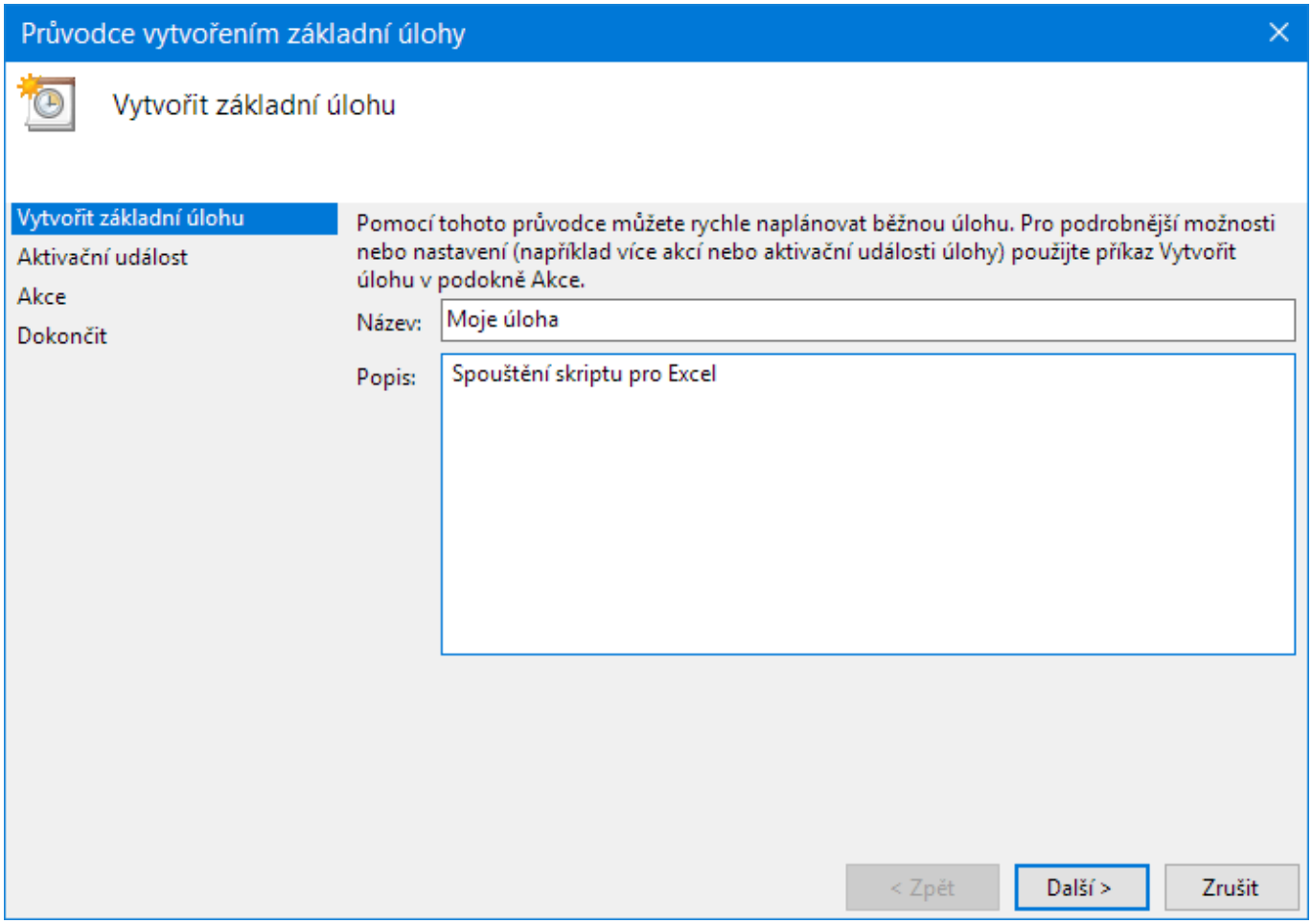
powershell -file „d:\skript_powershell.ps1“

V případě VBScriptu by se jednalo o příkaz

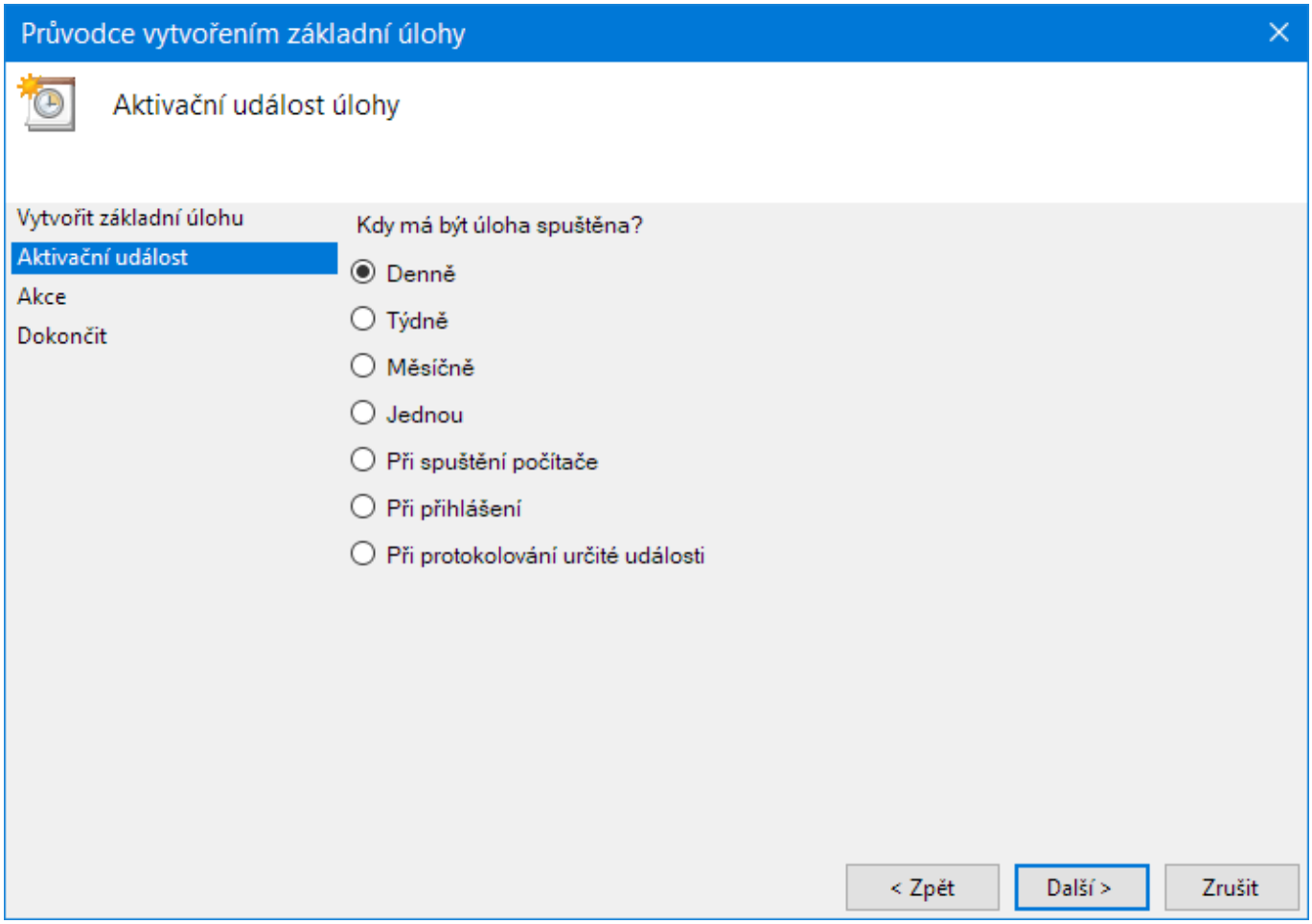
cscript skript_vbscript.vbs



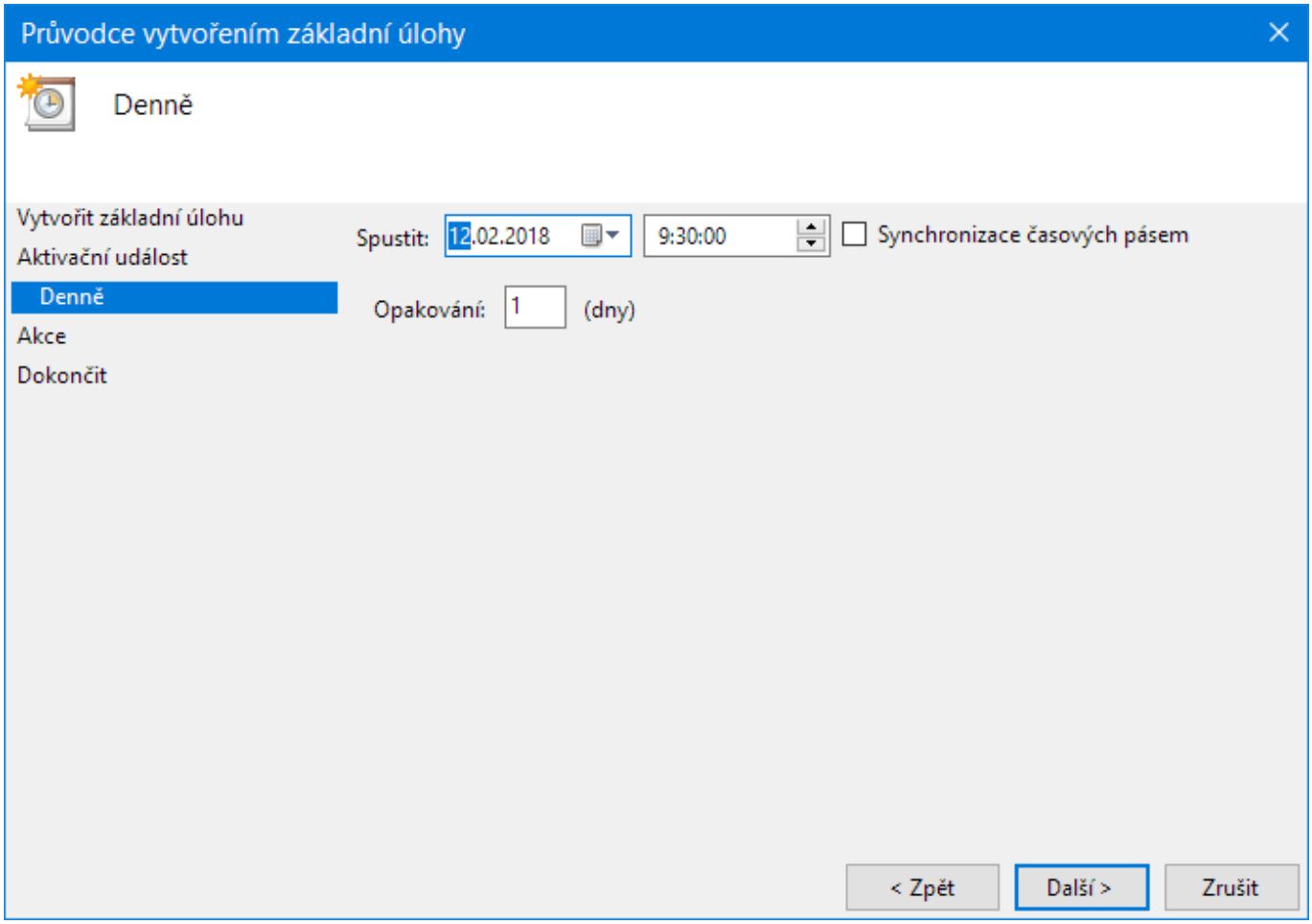

Plánovač úloh – krok 1



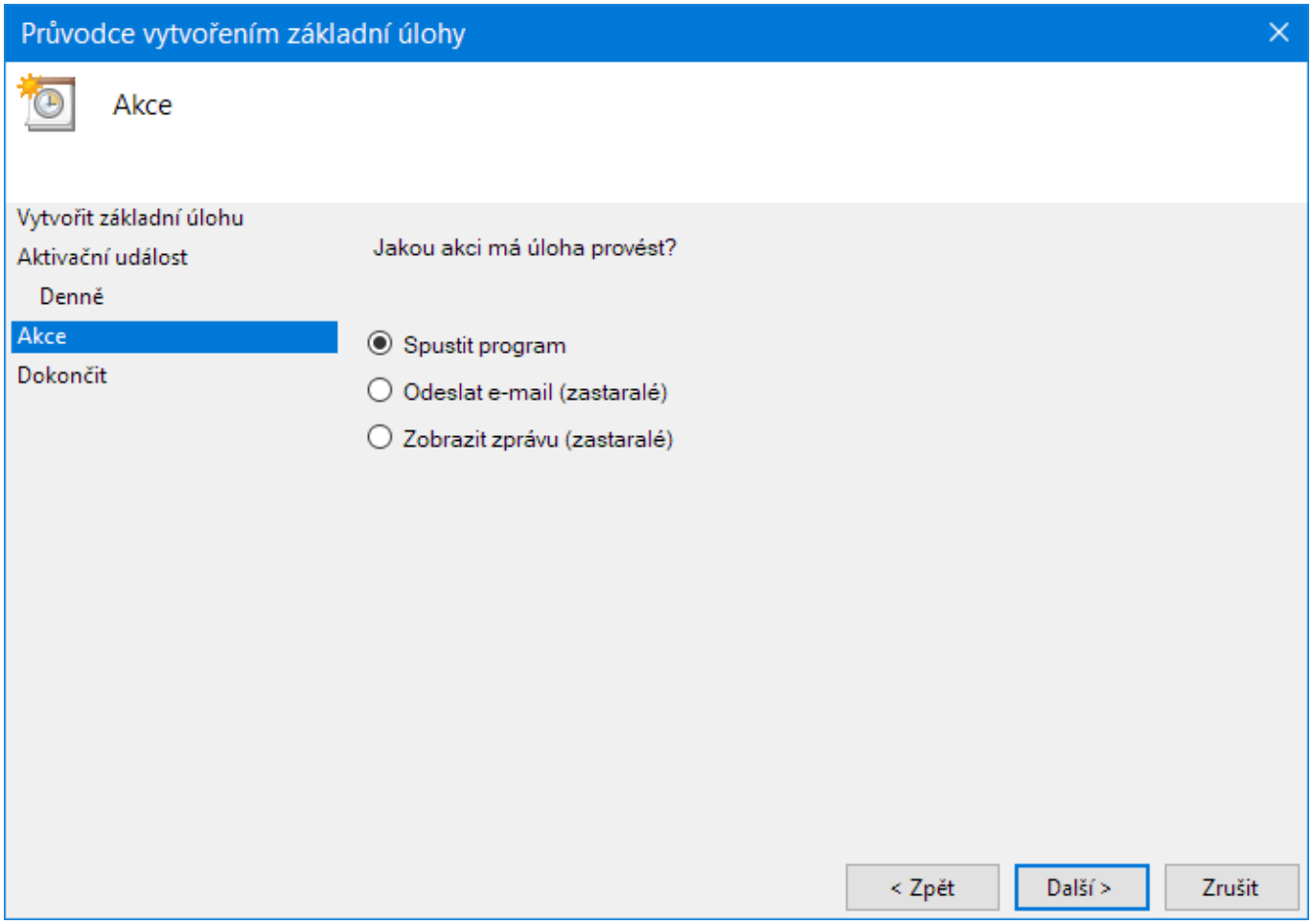

Plánovač úloh – krok 2



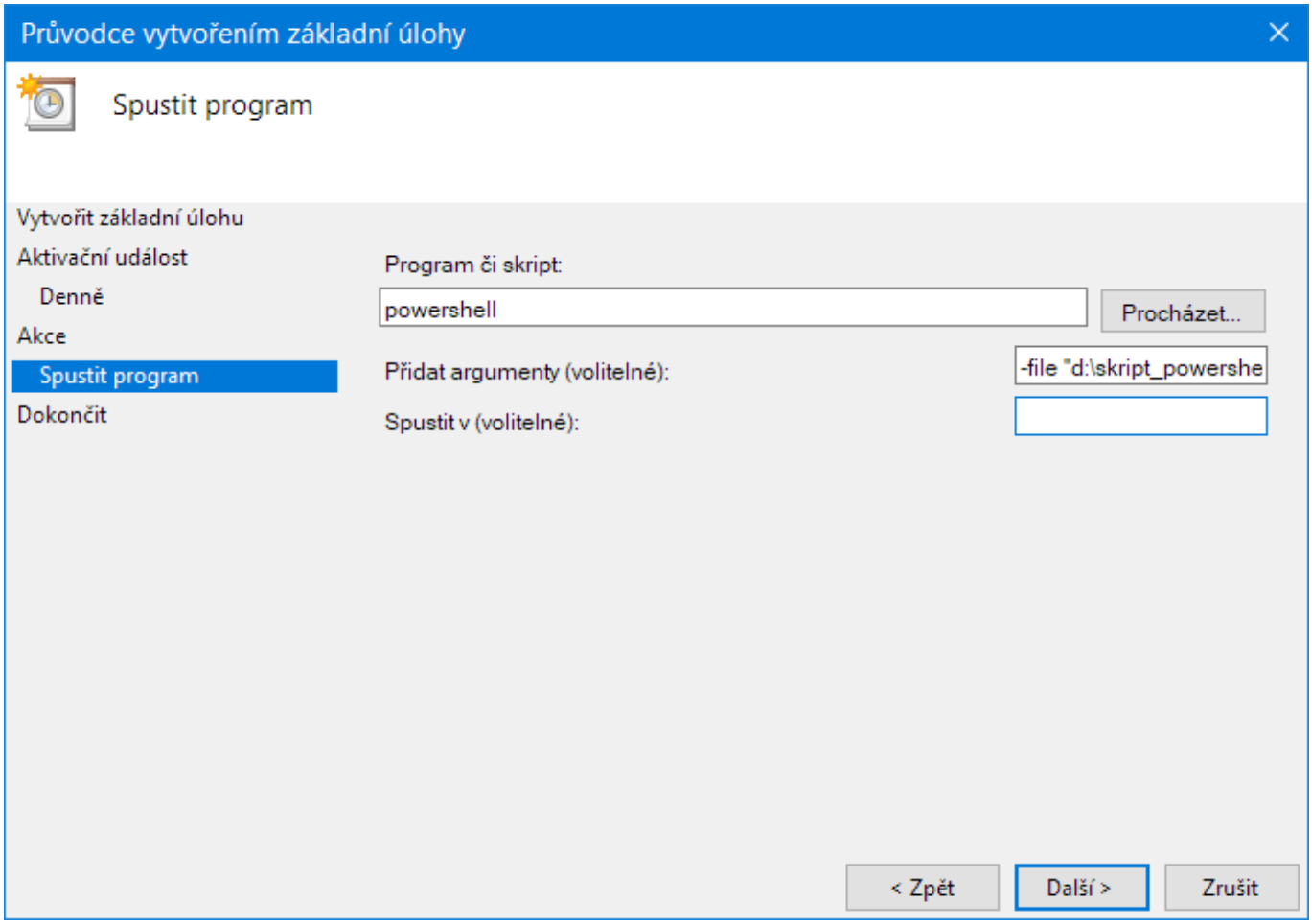

Plánovač úloh – krok 3



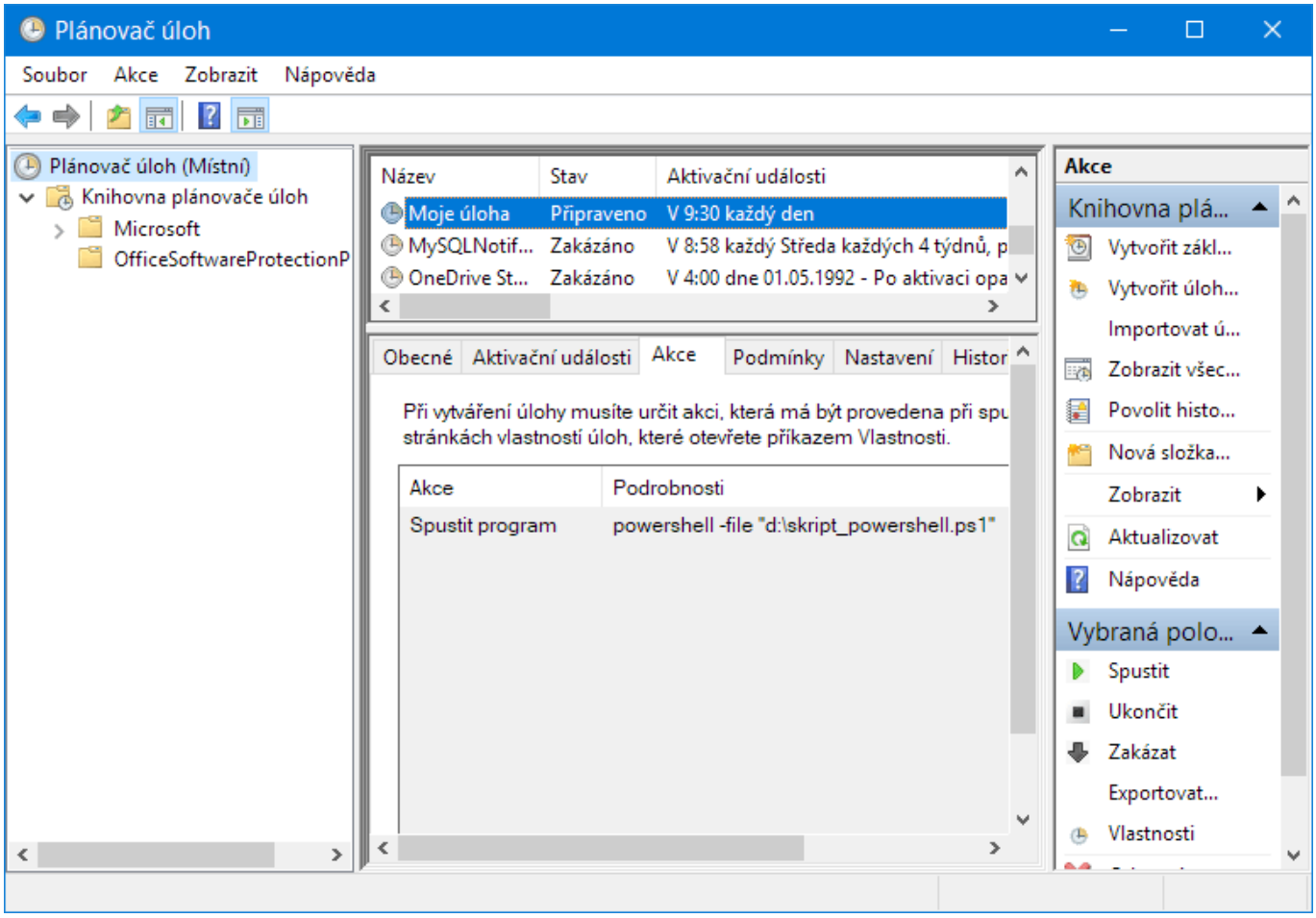

Plánovač úloh – krok 4




Plánovač úloh – krok 5




Plánovač úloh – krok 6




Plánovač úloh – krok 7

Ke stažení

excel_planovane_spusteni.zip