

12.3.1. opsi-template-with-userlogin

Eine andere, aufwendigere Möglichkeit dieses Problem zu lösen, ist temporär einen lokalen Benutzer anzulegen und diesen zur Installation des Programms zu verwenden. Dazu bieten wir als Vorlage das Produkt `opsi-template-with-userlogin` an. Dieses löst das bisher verwendete Produkt `opsi-template-with-admin` ab.

Wichtig

Verwenden Sie immer die aktuellste Version von `opsi-template-with-userlogin`!

Erzeugen eines angepassten Produkts

Um das Template an Ihre Bedürfnisse anzupassen empfiehlt sich das Erzeugen eines neuen Produktes auf Basis von `opsi-template-with-userlogin`:

```
opsi-package-manager -i --new-product-id myproduct opsi-template-with-userlogin_4.x.x.x-x.opsi
```

Ablauf

Das Produkt durchläuft die folgenden Schritte während der Installation:

- Sicherung der folgenden Werte:
 - Bisherige Auto Logon Einstellungen.
 - Zuletzt eingeloggter Benutzer.
 - User Account Control Einstellungen.
 - Hostparameter
`opsiclientd.event_software_on_demand.shutdown_warning_time`.
- Temporäres Setzen des Hostparameters
`opsiclientd.event_software_on_demand.shutdown_warning_time` auf den Wert "0", um unnötige Wartezeit zu vermeiden.
- Generieren eines nach konfigurierbaren Kriterien zufälligen Passwortes für den `opsiSetupUser`.
- Anlegen des lokalen Benutzers `opsiSetupUser`.
- Einrichten der Auto Logon Funktion für den Benutzer `opsiSetupUser`.
- Erstellen eines Scheduled Tasks für die Installation in der Aufgabenplanung.
- Je nach Einstellung der Product Property `execution_method` kopieren der Installationsdateien auf den Client.
- Neustart des Clients damit die Einstellungen für den Auto Logon in Kraft treten.
- Automatisches Einloggen des `opsiSetupUsers`.
- Ausführen der Installation über den angelegten Scheduled Task. Der Task startet mit einer Minute Verzögerung nach dem Login, damit alle Dienste genügend Zeit haben um zu starten.
- Abschließender Reboot des Clients.
- Aufräumen und Wiederherstellen des ursprünglichen Zustands.

- Löschen des opsiSetupUsers inklusive Benutzerprofils und Registry Einträgen.
- Löschen aller verwendeten lokalen Dateien.
- Wiederherstellen der gesicherten Werte von Auto Logon, zuletzt eingeloggtem Benutzer und der User Account Control.
- Wiederherstellen des ursprünglichen Werts des Hostparameters opsiClientd.event_software_on_demand.shutdown_warning_time.

Product Properties

Das Verhalten des Produkts kann über die folgenden Product Properties beeinflusst werden:

debug

- False (Default)
 - Sperrt Keyboard und Maus Eingaben während des Auto Logons des opsiSetupUsers um Benutzerinteraktion zu vermeiden. Das Passwort des opsiSetupUsers wird nicht im Klartext im Logfile angezeigt.
- True
 - Keyboard und Maus bleiben während des Auto Logons zum Debuggen im Fehlerfall aktiv. Das Passwort des opsiSetupUsers wird im Logfile im Klartext angezeigt.

execution_method

- event_starter_local_files
 - Während des Auto Logons wird die Installation über die opsiClientd_event_starter_asInvoker.exe getriggert, die den Server kontaktiert und dort ein on_demand Ereignis auslöst.
 - Die Installation wird im Kontext des System Users ausgeführt.
 - Der opsiSetupUser wird ohne Administrator Rechte angelegt.
 - Die Installationsdateien werden lokal auf dem Client gespeichert.
- event_starter_smb_share
 - Während des Auto Logons wird die Installation über die opsiClientd_event_starter_asInvoker.exe getriggert, die den Server kontaktiert und dort ein on_demand Ereignis auslöst.
 - Die Installation wird im Kontext des System Users ausgeführt.
 - Der opsiSetupUser wird ohne Administrator Rechte angelegt.
 - Die Installationsdateien liegen auf dem opsi_depot Share.
- local_winst_local_files (Default)
 - Während des Auto Logons erfolgt die Installation über den lokal installierten opsi-winst.
 - Die Installation wird im Kontext des opsiSetupUsers ausgeführt.

- Der opsiSetupUser wird mit Administrator Rechten angelegt.
- Die Installationsdateien werden lokal auf dem Client gespeichert.
- Befindet sich ein Client im WAN/VPN Modus (automatische Erkennung) wird diese Product Property ignoriert und die Installation mit folgenden Optionen durchgeführt:
 - Während des Auto Logons erfolgt die Installation über den lokal installierten opsi-winst.
 - Die Installation wird im Kontext des opsiSetupUsers ausgeführt.
 - Der opsiSetupUser wird mit Administrator Rechten angelegt.
 - Es werden die lokal im Cache vorhandenen Installationsdateien verwendet.

uninstall_before_install

- False (Default)
 - Vor der Installation wird keine Deinstallation ggf. bereits installierter Versionen durchgeführt.
- True
 - Vor der Installation wird geprüft ob die Software auf dem Client bereits vorhanden ist. Falls ja wird diese vor der Installation deinstalliert.

Aufbau des Produkts

Das Produkt gliedert sich in ein Hauptsript, das den Auto Logon und die Installation vorbereitet und das eigentliche Installationsscript, das während des Auto Logons des lokalen Benutzers ausgeführt wird.

===== Hauptsript Der besseren Übersicht halber verteilt sich das Hauptsript auf die folgenden Dateien:

- declarations.opsiinc (Auslagerung der Definition aller Variablen des Hauptscripts)
- sections.opsiinc (Auslagerung aller Sektionen des Hauptscripts)
- setup.opsiscript

Die einzigen Änderungen, die am Hauptsript vorgenommen werden müssen sind die Angabe des benötigten freien Speicherplatzes und die Parameter für die Generierung des für den Auto Login benötigten zufälligen Passwortes. Diese werden in der Datei `declarations.opsiinc` vorgenommen:

```

;-----
; - Please edit the following values -
;-----
;Available free disk space required
  Set $ProductSizeMB$ = "1000"

;Number of digits
  Set $RandomStrDigits$ = "3"

;Number of lower case characters
  Set $RandomStrLowerCases$ = "3"

;Minimum lenght of the generated string
  Set $RandomStrMinLength$ = "12"

;Number of special case characters
  Set $RandomStrSpecialChars$ = "3"

```

```

;Number of upper case characters
  Set $RandomStrUpperCases$ = "3"
; -----

```

=====
Installationsscript Auch das Installationsscript unterteilt sich der Übersicht halber auf mehrere Dateien:

- declarations-local.opsiinc (Auslagerung der Definition aller Variablen des Installationsscripts)
- sections-local.opsiinc (Auslagerung aller Sektionen des Installationsscripts)
- setup-local.opsiinc
- delsub-local.opsiinc
- uninstall-local.opsiinc

=====
Einfügen der Installationsdateien Öffnen Sie das Verzeichnis des Produktes in Ihrem Depot und legen Sie die Installationsdateien in das Unterverzeichnis `localsetup\files`. Die Dateien `Testfolder1` und `Testfile1.txt` können bedenkenlos gelöscht werden.

=====
Anpassen der Variablen Passen Sie die Variablen in der Datei `localsetup\declarations-local.opsiinc` an Ihre Bedürfnisse an:

```

; -----
; - Please edit the following values -
; -----
;The name of the software
  Set $ProductId$ = "opsi-template-with-userlogin"

;The folder that the software installs itself to
  Set $InstallDir$ = "%ProgramFilesSysNativeDir%" + $ProductId$

;Path to the installed executable
  Set $InstalledExecutable$ = $InstallDir$ + "\" + $ProductId$ +
".exe"

;Name of the license pool to be used
  Set $LicensePool$ = "p_" + $ProductId$

;Does the installation require a license?
  Set $LicenseRequired$ = "false"

;GUID of the installed MSI (Can be found in either HKLM\SOFTWARE\Microsoft
\Windows\CurrentVersion\Uninstall or HKLM\SOFTWARE\WOW6432Node\Microsoft
\Windows\CurrentVersion\Uninstall or determined by the opsi-setup-
detector)
  Set $MsiId$ = '{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}'

;Name of the uninstaller executable
  Set $Uninstaller$ = $InstallDir$ + "\uninstall.exe"
; -----

```

=====
Anpassen der Datei `setup-local.opsiinc` Die Installation der Software erfolgt über die Datei `setup-local.opsiinc` Diese beinhaltet neben dem Handling der Installation und des Lizenzmanagements auskommentierte Beispiele für das Kopieren von Dateien, das Erstellen von Registry Einträgen und das Anlegen von Verknüpfungen. Diese können, je nach Bedarf einkommentiert, gelöscht, oder auskommentiert gelassen werden.

=====
Anpassen der Datei `sections-local.opsiinc` Diese Datei beinhaltet alle von der Installation verwendeten Sektionen. In der Sektion `[Sub_Check_ExitCode]` muss die zum Installations Typ der verwendeten Software passende Funktion zur Auswertung des Exit Codes einkommentiert werden. Es können die Exit Codes für die folgenden Installations Typen ausgewertet werden:

- Inno Setup
- InstallShield
- MSI
- Nullsoft Scriptable Install System (NSIS)

Tip

Der Installations Typ kann über das Tool opsi-setup-detector ermittelt werden.

In diesem Beispiel wurde die Funktion `isMsiExitcodeFatal` einkommentiert:

```
[Sub_Check_ExitCode]
Set $ExitCode$ = getlastexitcode
;if stringtobool(isInnoExitcodeFatal($ExitCode$, "true", $ErrorString$ ))
;if stringtobool(isInstallshieldExitcodeFatal($ExitCode$, "true",
$ErrorString$ ))
if stringtobool(isMsiExitcodeFatal($ExitCode$, "true", $ErrorString$ ))
;if stringtobool(isNsisExitcodeFatal($ExitCode$, "true", $ErrorString$ ))
  Set $ErrorFlag$ = $ErrorString$
  Registry_Save Fatal_Flag /32Bit
  ExitWindows /ImmediateReboot
else
  Comment $ErrorString$
endif
```

Die Sektionen `Winbatch_Install` und `Winbatch_Uninstall` enthalten auskommentierte Beispiele für die Installations- bzw. Deinstallationskommandos der unterschiedlichen Installations Typen. Hier muss das passende Installations- bzw. Deinstallationskommando für den entsprechenden Installations Typ einkommentiert und angepasst werden.

```
[Winbatch_Install]
;Choose one of the following examples as basis for your installation
;You can use the variable $LicenseKey$ to pass a license key to the
installer

;===== Inno Setup =====
;"%ScriptPath%\localsetup\files\setup.exe" /sp- /silent /norestart

;===== InstallShield =====
;Create an setup.iss answer file by running: setup.exe /r
/f1"c:\setup.iss"
;"%ScriptPath%\localsetup\files\setup.exe" /s /sms
/f1"%ScriptPath%\localsetup\files\setup.iss"
/f2"$LogDir$\$ProductId$.install_log.txt"

;===== MSI package =====
;msiexec /i "%ScriptPath%\localsetup\files\setup.msi" /qb! /l*
"$LogDir$\$ProductId$.install_log.txt" ALLUSERS=1 REBOOT=ReallySuppress

;===== Nullsoft Scriptable Install System (NSIS) =====
;"%ScriptPath%\localsetup\files\setup.exe" /S <additional_parameters>

[Winbatch_Uninstall]
;Choose one of the following examples as basis for your uninstallation

;===== Inno Setup =====
;"$Uninstaller$" /silent /norestart

;===== InstallShield =====
;Create an uninstall.iss answer file by running: setup.exe /uninst /r
/f1"c:\uninstall.iss"
;"%ScriptPath%\localsetup\files\setup.exe" /uninst /s
/f1"%ScriptPath%\localsetup\files\uninstall.iss"
/f2"$LogDir$\$ProductId$.uninstall_log.txt"

;===== MSI =====
;msiexec /x $MsiId$ /qb! /l* "$LogDir$\$ProductId$.uninstall_log.txt"
REBOOT=ReallySuppress
```

```
;===== Nullsoft Scriptable Install System (NSIS) =====  
;"$Uninstaller$" /S
```

===== Anpassen der Datei `delsub-local.opsiinc` Beim Handling der Deinstallation wird entweder nach einer bereits installierten ausführbaren Datei, oder nach einer in der Registry vorhandenen MSI GUID gesucht. Hier muss, je nach Installations Typ die entsprechende Zeile ein- und die ander auskommentiert werden. Im folgenden Beispiel wurde die Zeile für MSI einkommentiert:

```
Comment "Searching for already installed version"  
;if FileExists($InstalledExecutable$)  
if NOT(GetRegistryStringValue("[HKLM\SOFTWARE\Microsoft\Windows  
\CurrentVersion\Uninstall\" + $MsiId$ + "] DisplayName") = "")  
  Comment "Starting the uninstallation"  
  Winbatch_Uninstall /SysNative  
  Sub_Check_ExitCode  
  
  Comment "License handling"  
  if NOT($LicenseRequired$ = "false")  
    Comment "Licensing required, free license used"  
    Sub_Free_License  
  endif  
  
  ;Comment "Deleting files"  
  ; Files_Delete /SysNative  
  
  ;Comment "Deleting registry entries"  
  ; Registry_Delete /SysNative  
  
  ;Comment "Deleting links"  
  ; LinkFolder_Delete  
endif
```

Analog zur Datei `setup-local.opsiinc` finden sich auch hier neben dem Handling der Deinstallation und des Lizenzmanagements auskommentierte Beispiele für das Löschen von Dateien, Registry Einträgen und Verknüpfungen. Diese können ebenfalls, je nach Bedarf einkommentiert, gelöscht, oder auskommentiert gelassen werden.

Anmerkung

Die Deinstallation erfolgt nicht im Kontext eines angemeldeten Benutzers, da dies meist nicht erforderlich ist.

Verhalten im Fehlerfall

Wichtig

Wird das Script angepasst muss dringend darauf geachtet werden dass die Funktion `isFatalError` in den lokalen Installationsscripten nicht verwendet wird! `isFatalError` bricht die Ausführung des Scripts **sofort** ab, was dazu führt dass die Cleanup Phase, in der ggf. die Tastatur und Maus Eingaben wieder aktiviert werden, die vorherigen Werte für z.B. den Auto Logon wiederhergestellt und der lokale `opsiSetupUser` gelöscht wird **nicht** ausgeführt wird! Das führt dazu dass der Client sich immer wieder als `opsiSetupUser` einloggt. Um dies zu unterbinden wird im Falle eines Fehlers die Fehlermeldung in der Variablen `$ErrorFlag$` abgelegt, in der Registry gespeichert und der Client per `ExitWindows /ImmediateReboot` sofort neu gestartet. Das führt dazu dass nach dem Reboot die Cleanup Phase durchlaufen, und dort der Fehler ausgewertet wird.

```
Set $ErrorFlag$ = "Installation not successful"
```

Registry_Save Fatal_Flag /32Bit
ExitWindows /ImmediateReboot