

Windows 10 Upgrade Dokumentation

Voraussetzungen

Die Clients, auf denen ein Upgrade ausgeführt werden soll, müssen den Windows 10 [Mindestvoraussetzungen](#) entsprechen.

Die Mindestvoraussetzungen für Windows 10 Release 2004 sind:

- RAM: 2GB of RAM for 64-bit and 1GB of RAM for 32-bit computers.
- Storage: 20GB of free space on 64-bit and 16GB of free space on 32-bit systems.
- Screen resolution: 800x600
- Graphics: Microsoft DirectX 9 or later with WDDM 1.0
- Intel CPU: All processors up 10th-gen, Intel Xeon E-22xx, Atom, Celeron, and Pentium.
- AMD CPU: All processors up to seventh-gen.
- Qualcomm CPU: Snapdragon 850 and 8cx.

Auf dem Client muss ein upgrade-fähiges Betriebssystem installiert sein. Die Windows 10 [Upgrade Matrix](#) bietet eine Übersicht der möglichen Upgrade Pfade.

Vorbereitungen

Download des Installationsmediums

Um ein Upgrade auf ein neueres Release durchführen zu können benötigt man ein entsprechendes Installationsmedium. Dabei müssen die folgenden Punkte beachtet werden:

1. Das Installationsmedium muss für die korrekte Architektur (32Bit/64Bit) heruntergeladen werden.
2. Das Installationsmedium muss für die korrekte Windows Edition (Home/Pro/Education/Enterprise) heruntergeladen werden.
3. Das Installationsmedium muss für die korrekte Sprache heruntergeladen werden.
4. Das Installationsmedium muss für die gewünschte Release ID (2004/20H2) heruntergeladen werden.

Installationsmedien können aus unterschiedlichen Quellen bezogen werden:

- [Microsoft Windows 10 Disc Image Download](#)
- [Media Creation Tool](#)
- [Volume Licensing Center](#)
- [MSDN](#)
- [Windows Insider Website](#)
- [Website for Education Editions \(Product Key required\)](#)

Integration des Installationsmediums

Um die benötigten Dateien zu erhalten muss die heruntergeladene ISO Datei entpackt werden. Die entpackten Dateien müssen in den Ordner `localsetup\installfiles` des windows10-upgrade Pakets kopiert werden. Danach sollte ein `sudo opsi-set-rights` ausgeführt werden. Es können auch mehrere Installationsmedien in das Paket integriert werden um beispielsweise auf verschiedene Release IDs upgraden zu können, oder verschiedene Architekturen oder Sprachen abzudecken. Dabei ist folgendes zu beachten:

- Der Ordnername muss immer mit `installfiles` beginnen!
- Um die Product Property automode verwenden zu können muss die Release ID Teil des Ordnernames sein!

Beispiele:

- `installfiles2004`
- `installfiles-20h2`
- `installfiles_21h1-32Bit` (ab Version 20.09-3)
- `installfiles_21h2-en` (ab Version 20.09-3)

Damit das Paket die verschiedenen Installfiles Ordner erkennt, muss es nach dem Anlegen und Befüllen der Ordner per `opsi-package-manager -i` erneut installiert werden, da die Integration der Installfiles Ordner in den `preinst/postinst` Scripten erfolgt. Das zu verwendende Installfiles Verzeichnis kann danach über die Product Property `installfiles_dir` ausgewählt werden.

Überprüfen des Installationsmediums

Mit den Tools `wiminfo` oder `wimlib` können die in der `install.wim` enthaltenen Images ausgelesen werden. Damit können die im Installationsmedium enthaltenen Editionen, Sprachen und die Architektur ermittelt werden.

wiminfo

Das Tool wird unter Linux über das Server Paket `opsi-windows-support` bereit gestellt. Mit `wiminfo` können die Informationen direkt auf der Konsole des OPSI Servers ausgelesen werden. Beispiel Befehl zum Auslesen der Informationen:

- `wiminfo /var/lib/opsi/depot/windows10-upgrade/localsetup/installfiles/sources/install.wim`

wimlib

Das Tool liegt im Paket unter den folgenden Pfaden:

- `localsetup\wimlib\x64\wimlib-imagex.exe`
- `localsetup\wimlib\x86\wimlib-imagex.exe`

Beispiel Befehl zum Auslesen der Informationen (\\<opsi-server>\opsi_depot Freigabe als Laufwerk p:\ gemounted):

- `p:\windows10-upgrade\localsetup\wimlib\x64\wimlib-imagex.exe" info "p:\windows10-upgrade\localsetup\installfiles2009\sources\install.wim"`

Während des Upgrades wird das im per Product Property `installfiles_dir` ausgewählten Verzeichnis enthaltene Image mit `wimlib` ausgelesen und die Ausgabe im Logfile gespeichert.

Symlinks

Ab Version 21h1-1 müssen die Installfiles Ordner nicht direkt im Ordner `localfiles` abgelegt werden, sondern können auch per Symlink eingebunden werden. Beispiel zum Anlegen eines Symlinks auf den Installfiles Order aus dem netboot Paket `win10-x64`:

- `cd /var/lib/opsi/depot/windows10-upgrade/localsetup`
- `ln -s ../../win10-x64/installfiles/ .`

Nach dem Anlegen des symlinks muss das Paket per `opsi-package-manager -i` erneut installiert werden, da die Integration der Installfiles Ordner in den `preinst/postinst` Scripten erfolgt. Das zu verwendende Installfiles Verzeichnis kann danach über die Product Property `installfiles_dir` ausgewählt werden.

Product Properties

- `automode`
 - `False` (Default)
 - Die Release ID, auf die das Upgrade durchgeführt werden soll, muss über die Product Property `upgrade_to_version` ausgewählt werden.
 - `True`
 - Es wird versucht die Release ID aus dem Ordernamen des über die Product Property `installfiles_dir` ausgewählten Ordners auszulesen.
 - Beispiel: Wurde in der Product Property `installfiles_dir` der Ordner `installfiles2004` ausgewählt, wird 2004 als Release ID verwendet. Die unter `upgrade_to_version` ausgewählte Release ID wird in diesem Fall ignoriert!
 - Falls keine gültige Release ID ausgelesen werden kann, wird der unter der Product Property `upgrade_to_version` ausgewählte Wert als Fallback verwendet.
- `bitlocker_suspend`
 - `AlwaysSuspend` (Default)
 - BitLocker wird während eines Upgrades immer angehalten.
 - `ForceKeepActive`
 - Ermöglicht Upgrades ohne Anhalten von BitLocker. Wenn das Upgrade nicht abgeschlossen werden kann, weil BitLocker aktiv ist, schlägt das Upgrade fehl.
 - `TryKeepActive`
 - Versucht ein Upgrade, ohne BitLocker anzuhalten. Wenn das Upgrade fehlschlägt, hält Windows Setup BitLocker an und schließt das Upgrade ab. Dies ist das Standardverhalten, wenn die Option `/BitLocker` nicht angegeben ist.
- `copy_files_locally`
 - `False` (Default)
 - Die Installationsdateien werden während des Upgrades vom `opsi_depot` bezogen
 - `True`

- Die Installationsdateien werden vor dem Upgrade lokal in den Ordner %SystemDrive%\opsi.org\usertmp\windows10-upgrade kopiert
- debug
 - False (Default)
 - Sperrt Keyboard und Maus Eingaben während des Auto Logons des opsiSetupUsers um Benutzerinteraktion zu vermeiden.
 - True
 - Keyboard und Maus bleiben während des Auto Logons zum Debuggen im Fehlerfall aktiv.
- delete_windows_old
 - False (Default)
 - Der Windows.old Ordner bleibt nach dem Upgrade erhalten.
 - True
 - Nach einem erfolgreichen Upgrade wird der Windows.old Ordner gelöscht.
- delete_windows_update_cache
 - False (Default)
 - Der Windows Update Cache im Ordner SystemRoot\SoftwareDistribution bleibt erhalten
 - True
 - Der Windows Update Cache im Ordner SystemRoot\SoftwareDistribution wird vor dem Upgrade gelöscht
- dynamic_update
 - Disable (Default)
 - Es werden keine dynamische Aktualisierungsvorgänge durchgeführt.
 - Enable
 - Es werden dynamische Aktualisierungsvorgänge durchgeführt.
 - NoDrivers
 - Dynamische Aktualisierungsvorgänge werden mit Ausnahme der Treibererfassung durchgeführt.
 - NoDriversNoLCU
 - Dynamische Aktualisierungsvorgänge werden durchgeführt, mit Ausnahme von Treibern und der letzten kumulativen Aktualisierung.
 - NoLCU
 - Dynamische Update-Operationen werden durchgeführt, mit Ausnahme des letzten kumulativen Update-Erwerbs.
- encryption_driver
 - Angabe eines Pfades der Treiber für Third Party Verschlüsselungs Software enthält.
- execution_method
 - loginOpsiSetupUser
 - Der opsiClientD Dienst erzeugt den User opsiSetupUser und loggt diesen ein.
 - Als Shell wird die explorer.exe gestartet.
 - Der Desktop des opsiSetupUsers ist sichtbar.
 - Nach dem Einloggen startet nach einer kurzen Verzögerung das Upgrade.
 - Die Installationsdateien können vom Server abgerufen, oder lokal auf dem Client gespeichert werden (siehe copy_files_locally).
 - runAsOpsiSetupUser
 - Der opsiClientD Dienst erzeugt den User opsiSetupUser und loggt diesen ein.
 - Als Shell wird die powershell.exe gestartet und durch diese der lokale opsi-script getriggert, welcher das Upgrade ausführt.
 - Der Desktop des opsiSetupUsers ist nicht sichtbar.
 - Die Installationsdateien müssen lokal auf dem Client gespeichert werden.

- Der unter `copy_files_locally` eingestellte Wert wird ignoriert und automatisch auf `true` gesetzt.
- `runOpsiScriptAsOpsiSetupUser` (default)
 - Der `opsiclientd` Dienst erzeugt den User `opsiSetupUser` und loggt diesen ein.
 - Als Shell wird die `powershell.exe` gestartet und durch diese die `action_processor_starter.exe` getriggert, welche eine Service Connection aufbaut und danach das Upgrade ausführt.
 - Der Desktop des `opsiSetupUsers` ist nicht sichtbar.
 - Die Installationsdateien können vom Server abgerufen, oder lokal auf dem Client gespeichert werden (siehe `copy_files_locally`).
- Befindet sich ein Client im WAN/VPN Modus (automatische Erkennung) wird diese Product Property ignoriert und die Installation mit folgenden Optionen durchgeführt: (Siehe auch Kapitel WAN/VPN Modus)
 - Der `opsiclientd` Dienst erzeugt den User `opsiSetupUser` und loggt diesen ein.
 - Als Shell wird die `powershell.exe` gestartet und durch diese der lokale `opsi-script` getriggert, welcher das Upgrade ausführt.
 - Der Desktop des `opsiSetupUsers` ist nicht sichtbar.
 - Es werden die lokal im Cache vorhandenen Installationsdateien verwendet.
- `installfiles_dir`
 - Auswahl des Ordners, der die Installationsdateien enthält.
 - Der Ordnername muss immer mit `installfiles` beginnen!
 - Soll die Product Property `automode` verwendet werden, muss die Release ID an den Ordnernamen angehängt werden (z.B. `installfiles2004`)
 - Damit alle im Paket enthaltenen Installfiles Verzeichnisse in der Product Property `installfiles_dir` aufgelistet werden muss ggf. nach dem Anlegen der Ordner das Paket erneut per `opsi-package-manager -i` installiert werden, da die Property über die `preinst/postinst` Skripte befüllt wird.
- `image_index`
 - Leer (Default)
 - Weist Windows Setup an, welches Betriebssystem-Image von `install.wim` installiert werden soll, wenn möglicherweise mehrere Images anwendbar sind.
- `language_from_language_pack` (Siehe Kapitel Language Packs)
 - `False` (default)
 - Auswählen falls der Client nicht mit einem Installationsmedium installiert wurde das Language Packs enthält.
 - `True`
 - Auswählen falls der Client mit einem Installationsmedium installiert wurde das Language Packs enthält und die Sprache des Clients von einem Language Pack bereit gestellt wird.
- `language_installation_media` (Siehe Kapitel Language Packs)
 - Angabe des Language Codes des originalen Installationsmedium des Clients
 - Beispiele:
 - 0407 für deutsch
 - 0409 für englisch
- `migrate_drivers`
 - `All` (Default)
 - Weist Windows Setup an, alle Treiber während des Upgrades aus der vorhandenen Installation zu migrieren.
 - `None`
 - Weist Windows Setup an, keine Treiber während des Upgrades aus der vorhandenen Installation zu migrieren.

- mode
 - upgrade (Default)
 - Upgrade auf die ausgewählte Release ID.
 - validate_before_upgrade
 - Trial Run um Kompatibilitäts Probleme zu ermitteln. Werden keine Probleme ermittelt wird das Upgrade gestartet.
 - validate_only
 - Trial Run um Kompatibilitäts Probleme zu ermitteln.
- post_oobe
 - Pfad zu setupcomplete.cmd
- Weitere Infos unter:
 - <https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/add-a-custom-script-to-windows-setup?view=windows-11>
- productkey
 - Lizenz Key, der für die Installation verwendet werden soll.
- proxy_ip
 - Ermöglicht die Angabe der IP eines Proxy Servers.
- proxy_port
 - Ermöglicht die Angabe der Portnummer eines Proxy Servers.
- quiet
 - False
 - Die Installation startet im interaktiven Modus und erfordert die Eingaben eines Benutzers.
 - Nützlich um im Fehlerfall eine aussagekräftige Fehlermeldung zu erhalten.
 - True (Default)
 - Die Installation startet im silent Modus ohne Benutzerinteraktion.
- restore_health
 - False (default)
 - „SFC /scannow“ and „DISM /online /cleanup-image /restorehealth“ werden vor dem Upgrade nicht ausgeführt
 - True
 - „SFC /scannow“ and „DISM /online /cleanup-image /restorehealth“ werden vor dem Upgrade ausgeführt.
- setup_after_install
 - Auswahl der OPSI Produkte, die nach einem erfolgreichen Upgrade installiert werden sollen.
 - Auswählbare Produkte (manuell erweiterbar):
 - config-win10
 - mshotfix
 - opsi-client-agent
- showoobe
 - Full
 - Benutzer müssen sich beim Einloggen nach dem Upgrade durch den Out Of Box Experience (OOBE) Dialog klicken.
 - None (Default)
 - Überspringt den Out Of Box Experience Dialog (OOBE) beim Einloggen eines Benutzer nach dem Upgrade.
- telemetry
 - Disable (Default)
 - Es werden keine Telemetrie Daten des Upgrades an Microsoft übermittelt.
 - Enable

- Telemetrie Daten des Upgrades werden an Microsoft übermittelt.
- `timeout_opsisetupuser`
 - 120 (Default)
 - Gibt den Timeout Wert in Minuten für `opsiServiceCallrunAsOpsiSetupUser` und `opsiServiceCallrunOpsiScriptAsOpsiSetupUser` an
- `unset_after_install`
 - Auswahl der OPSI Produkte, die nach einem erfolgreichen Upgrade auf `not_installed` gesetzt werden sollen.
- `upgrade_to_version`
 - Auswahl der Release ID, auf die das Upgrade ausgeführt werden soll.
 - Wird ignoriert, falls `automode` auf `true` steht.
 - Nur falls per Automode keine gültige Release ID ausgelesen werden konnte, wird der hier ausgewählte Wert als Fallback verwendet.
 - `latest` (Default)
 - Verweist immer auf die aktuellste Release ID, die zum Zeitpunkt der Paketerstellung aktuell war.

WAN/VPN Modus

Das Paket erkennt automatisch, ob sich ein Client im WAN/VPN Modus befindet. Es werden die lokal gecachten Installationsdateien verwendet.

Besonderheiten

Wichtig! Nach jeder Änderung an den Installfiles Ordnern muss das Paket erneut per `opsi-package-manager -i` installiert werden, damit die `.files` Datei neu generiert wird!

Pakete, die per `setup_after_install` ausgewählt wurden, können erst gecached werden wenn der Client das nächste Mal erfolgreich den Server kontaktieren kann. Meist geschieht das erst durch eine aufgebaute VPN Verbindung nach dem Login eines Users.

Language Packs

Windows Installationen mit installierten Language Packs stellen einen Sonderfall dar. Um eine Installation die Language Packs verwendet erfolgreich zu aktualisieren müssen einige Punkte beachtet werden.

Vorbereitung des Installationsmediums

Alle benötigten Language Packs müssen in die `install.wim` des Installationsmediums für das Upgrade eingebunden werden. Dabei ist zu beachten dass die Language Packs passend zur Version des Installationsmediums heruntergeladen werden. Language Packs in Version 1909 können beispielsweise nicht in ein 21H1 Installationsmedium eingebunden werden. Das Einbinden der Language Packs erfolgt über das Tool DISM. Anleitungen und weitere Informationen finden sich in den folgenden Links:

[Add or Remove Packages Offline Using DISM](#)
[How to Add Language Packs Offline Using DISM](#)
[DISM Languages and International Servicing Command-Line Options](#)
[Available language packs for windows](#)

Ablauf

Das zu aktualisierende Betriebssystem muss vor dem Upgrade auf die Sprache des originalen Installationsmediums zurück gestellt werden.

Wurde ein Client beispielsweise von einem englischen Installationsmedium installiert, das ein deutsches Language Pack enthält und Deutsch wurde als Hauptsprache des Clients konfiguriert muss die Sprache vor dem Upgrade auf englisch eingestellt werden.

Nach dem Upgrade kann die Sprache wieder auf deutsch zurück gestellt werden, sofern in das beim Upgrade verwendete Installationsmedium ein zu dessen ReleaseID passendes deutsches Language Pack integriert wurde.

Dieser Vorgang wurde in der Version 21h1-1 der windows-upgrade Pakets automatisiert eingebunden. Vor dem Upgrade wird die derzeitige Sprache ausgelesen und gespeichert. Dann wird die Sprache auf die Sprache des zur Installation des Clients verwendeten Installationsmedium eingestellt. Nach dem Upgrade wird die Sprache wieder auf die anfangs ausgelesenen Werte zurück gestellt.

Client Einstellungen

- Die Product Property `language_from_language_pack` muss auf `true` gestellt werden.
- Die Product Property `language_installation_media` muss auf den vierstelligen Language Code des Installationsmediums eingestellt werden.
 - Beispiele:
 - 0407 für deutsch
 - 0409 für englisch

Upgrade von Windows 10 Vorgängerversionen

Die folgenden Windows 10 Vorgängerversionen können auf Windows 10 aktualisiert werden:

- Windows 7
- Windows 8
- Windows 8.1

Siehe auch: [Windows Upgrade Matrix](#)

Vor dem Upgrade sollten alle aktuellen Windows Updates eingespielt werden.

Debugging

Eine Vielzahl von Umständen kann dazu führen dass ein Upgrade fehlschlägt:

- Client entspricht nicht den Mindestvoraussetzungen für Windows 10

- System Reserved Partition zu klein
- Virens Scanner verhindert das Upgrade
- Inkompatible Treiber auf dem Client
- Inkompatible Software auf dem Client
- Falsches Installationsmedium
 - Falsche Architektur
 - Falsche Sprache
 - Falsche Edition
- Aktivierungsstatus des Betriebssystems vor dem Upgrade
- Benutzerinteraktion während des Upgrades (z.B. Neustart oder Ausschalten des Clients)

Auswertung des Exit Codes

Ab Version 20.09-1 wird der Exit Codes des Upgrades detaillierter ausgewertet und eine aussagekräftige Fehlermeldung mit Tipps zur Behebung in das Logfile geschrieben. Die Liste der Exit Codes ist aber mit Sicherheit nicht komplett, da es keine vollständige Auflistung aller Exit Codes seitens Microsoft gibt.

Inkludierte Upgrade Logfiles

Die folgenden Upgrade Logfiles werden (sofern vorhanden) in das Logfile des Pakets eingebunden und können bequem im Configed analysiert werden:

- „%opsiLogDir%\windows10-upgrade.log\Panther*APPRAISER*.xml“
- „%opsiLogDir%\windows10-upgrade.log\Panther*CompatData*.xml“
- „%opsiLogDir%\windows10-upgrade.log\MoSetup\ActionList.xml“
- „%opsiLogDir%\windows10-upgrade.log\MoSetup\Bluebox.log“
- „%opsiLogDir%\windows10-upgrade.log\MoSetup\DeviceInventory.xml“
- „%SystemRoot%\Logs\SetupDiag\setupdiagresults.xml“
- „%opsiLogDir%\opsisetupuser.log“
- „%opsiLogDir%\SetupDiagResults.log“
- „%opsiLogDir%\windows10-upgrade.log\Panther\diagerr.xml“
- „%opsiLogDir%\windows10-upgrade.log\Panther\MigLog.xml“
- „%opsiLogDir%\windows10-upgrade.log\Panther\ScanResult.xml“
- „%opsiLogDir%\windows10-upgrade.log\Panther\setupact.log“
- „%opsiLogDir%\windows10-upgrade.log\Panther\setuperr.log“
- „%opsiLogDir%\sfcdetails.txt“
- „%SystemRoot%\logs\dism\dism.log“

Lösungsansätze für Probleme mit dem Windows Upgrade

Um inkompatible Software und Treiber zu ermitteln sucht man im windows10-upgrade Logfile, oder direkt in „%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\scanresult.xml“ und %SystemDrive%\\$WINDOWS.~BT\Sources\Panther\Compatdata*.xml (mehrere einzelene Logdateien) nach den folgenden Strings:

- CompatibilityInfo BlockingType="Hard"
- CompatibilityInfo BlockMigration="True"

Im Fall eines inkompatiblen Treibers sollte man die entsprechende .inf Datei angezeigt bekommen (z.B oem11.inf). Die .inf Dateien liegen im Ordner %SystemDrive%\Windows\INF. Diese öffnet man mit einem Texteditor um den Treibernamen zu ermitteln. Den, oder die gefundenen Treiber deinstalliert man auf dem Client und startet das Upgrade erneut.

Achtung! Es werden **immer** zwei Treiber mit dem Status CompatibilityInfo BlockMigration="True" gefunden (je nach System in unterschiedlichen oemxx.inf Dateien) Die zugehörigen Geräte lauten:

- Microsoft Print to PDF
- Microsoft XPS Document writer

Obwohl diese beiden Treiber mit CompatibilityInfo BlockMigration="True" angezeigt werden, verhindern diese **nicht** das Upgrade und können ignoriert werden.

Um eine aussagekräftige Fehlermeldung zu erhalten kann man das Upgrade mit den Product Properties quiet = false und debug = true starten. **Achtung!** Dadurch startet das Upgrade im interaktiven Modus und erfordert Benutzereingaben! Während des Upgrade Vorgangs erhält man **meistens** in der GUI eine genauere Fehlermeldung.

Das Upgrade kann mit der Product Property mode = validate oder mode = validate_before_upgrade gestartet werden. In diesem Fall wird lediglich ein Testlauf des Upgrades gestartet. Eventuell auftretende Fehler können somit vor dem eigentlichen Upgrade ermittelt werden.

Lösungsansätze für Probleme mit dem windows10-upgrade Paket

Es kann leider immer wieder auch zu Problemen mit dem windows10-upgrade Paket selbst kommen. Generell empfiehlt es sich immer die aktuellste Version des windows10-upgrade Pakets zu verwenden!

Auto Logon Probleme

Es können Probleme mit dem Auto Logon des opsiSetupUsers auftreten. Meist verhindern Gruppenrichtlinien dass sich der temporäre opsiSetupUser einloggen darf. In diesem Fall sollte überprüft werden ob der Logon nur für bestimmte lokale User erlaubt ist, oder ob vielleicht die Passwort Richtlinie den Login verhindert.

Windows Dialoge, die beim Login eine Benutzerinteraktion erfordern können das automatische Einloggen ebenfalls verhindern. Das windows10-upgrade Paket unterdrückt alle bekannten Dialoge, es können je nach Release ID aber auch immer neue Dialoge hinzukommen, die das Einloggen verhindern. In diesem Fall wird immer ein Screenshot des entsprechenden Dialogs benötigt, damit ein Workaround dafür in die nächste Version des windows10-upgrade Pakets eingebaut werden kann.