

# Windows 11 Upgrade Documentation

## Prerequisites

The clients on which an upgrade is to be performed must meet the Windows 11 [Minimum Prerequisites](#). The minimum requirements for Windows 11 Release 22H2 are:

- Processors/CPU (Central Processing Units):
  - 1 GHz or faster with 2 or more cores listed on our list of approved CPUs. The processor in your PC is a critical factor in running Windows 11. Clock speed (1 GHz or faster) and number of cores (2 or more) are inherent with the processor design as manufactured and are not considered upgradeable components.
- RAM:
  - 4 GB. If your PC has less than 4 GB of RAM, there is sometimes an option to upgrade additional RAM. Check with your PC manufacturer's website or a retailer to see if there are easy and inexpensive options to meet the minimum Windows 11 requirements.
- Memory:
  - 64 GB or larger storage device. If your PC does not have a large enough storage drive, there are sometimes ways to upgrade the drive. Check with your PC manufacturer's website or a reseller to see if there are easy and inexpensive options to meet the minimum requirements for Windows 11.
- System firmware:
  - UEFI (for Unified Extensible Firmware Interface, a modern version of the PC BIOS) and Secure Boot-enabled. If your device does not meet the minimum requirements because it is not Secure Boot-enabled, you should read this article to see if there are steps you can take to enable it. Secure Boot can only be enabled with UEFI, and this article will help you understand the possible options to change the settings to enable this.
- TPM:
  - Trusted Platform Module (TPM) version 2.0, if your device does not meet the minimum requirements because of the TPM, you should read this article to see if there are steps you can take to fix this.
- Graphics card:
  - Compatible with DirectX 12 or higher with WDDM 2.0 driver.
- Display:
  - High resolution screen (720p) with a diagonal greater than 9 inches and 8 bits per color channel. If your screen is smaller than 9,, the intended Windows user interface may not be fully visible.
- Internet connection and Microsoft accounts:
  - Windows 11 Home Edition requires an Internet connection and a Microsoft account to complete device setup when first used.
- Windows version for upgrade:
  - Your device must be running Windows 10, version 2004 or later, to upgrade via Windows Update. Free updates are available through Windows Update under Settings > Update and Security.

# Preparations

## Download the installation media

In order to upgrade to a newer version, you need an appropriate installation medium. The following points must be observed:

1. The installation media must be downloaded for the correct architecture (32bit/64bit).
2. The installation media must be downloaded for the correct Windows edition (Home/Pro/Education/Enterprise).
3. The installation media must be downloaded for the correct language.
4. The installation media must be downloaded for the correct Release ID (2004/20H2).

Installation media can be obtained from various sources:

- [Download Windows 11](#)
- [Volume Licensing Center](#)
- [MSDN](#)
- [Windows Insider Website](#)

## Integration of the installation media

To obtain the required files the downloaded ISO file must be unzipped. The unzipped files must be copied to the `localsetup\installfiles` folder of the Windows 11 upgrade package. After that a `sudo opsi-set-rights` should be executed. Multiple installation media can also be included in the package, for example to upgrade to different release IDs, or to cover different architectures or languages. The following should be noted:

- The folder name must always begin with installation files!
- To use the product property automode the release ID must be part of the folder name!

Examples:

- `installfiles22h2`
- `installfiles-22h2`
- `installfiles_22h2-32Bit`
- `installfiles_22h2-en`

In order for the package to recognize the different Installfiles folders, it must be reinstalled via `opsi-package-manager -i` after the folders have been created and filled, since the integration of the Installfiles folders is done in the `preinst/postinst` scripts. The installfiles directory to be used can then be selected via the `installfiles_dir` product property.

## Checking the installation media

With the tools `wiminfo` or `wimlib` the images contained in the `install.wim` can be read out. This can be used to determine the editions, languages and architecture contained in the installation medium.

## wiminfo

The tool is provided on Linux via the server package `opsi-windows-support`. With `wiminfo` the information can be read out directly on the console of the OPSI server.\ Example command to read the information:

- `wiminfo /var/lib/opsi/depot/windows11-upgrade/localsetup/installfiles/sources/install.wim`

## wimlib

The tool is located in the package under the following paths:

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

Example command to read the information (`\\<opsi-server>\opsi_depot` share mounted as drive `p:`):

- `p:\windows11-upgrade\localsetup\wimlib\x64\wimlib-imagex.exe" info "p:\windows11-upgrade\localsetup\installfiles22h2\sources\install.wim".`

During the upgrade, the image contained in the directory selected by product property `installfiles_dir` is read with `wimlib` and the output is stored in the logfile.

## Symlinks

As of version 21h1-1, the `installfiles` folders do not have to be stored directly in the `localfiles` folder, but can also be included via symlink. Example for creating a symlink to the `installfiles` folder from the netboot package `win11-x64`:

- `cd /var/lib/opsi/depot/windows11-upgrade/localsetup`
- `ln -s ../../win11-x64/installfiles/ .`

After creating the symlink, the package must be installed again via `opsi-package-manager -i`, because the integration of the `installfiles` folders is done in the `preinst/postinst` scripts. The `installfiles` directory to be used can then be selected via the `installfiles_dir` product property.

## Product Properties

- `automode`
  - False (Default)
    - The release ID to upgrade to must be selected via the product property `upgrade_to_version`.
  - True
    - An attempt is made to read the release ID from the folder name of the folder selected via the `installfiles_dir` Product Property.
    - Example: If the folder `installfiles22h2` was selected in the `installfiles_dir`

product property, 22h2 will be used as the release ID. The release ID selected under `upgrade_to_version` will be ignored in this case!

- If no valid release ID can be read, the value selected under the `upgrade_to_version` product property is used as fallback.

- `copy_files_locally`

- False (Default)
  - The installation files are retrieved from `opsi_depot` during upgrade
- True
  - The installation files are copied locally to the folder `%SystemDrive%\opsi.org\usertmp\windows11-upgrade` before upgrading

- `debug`

- False (Default)
  - Locks keyboard and mouse input during `opsiSetupUsers` auto logon to avoid user interaction.
- True
  - Keyboard and mouse remain active during Auto Logon for debugging in case of error.

- `delete_windows_old`

- False (Default)
  - The `Windows.old` folder is kept after the upgrade.
- True
  - After a successful upgrade, the `Windows.old` folder is deleted.

- `delete_windows_update_cache`

- False (Default)
  - The Windows Update cache in the folder `SystemRoot\SoftwareDistribution` is kept.
- True
  - The Windows Update cache in the `SystemRoot\SoftwareDistribution` folder is cleared before the upgrade.

- `encryption_driver`

- Specifies a path that contains drivers for third-party encryption software.

- `execution_method`

- `loginOpsiSetupUser`
  - The `opsiclientd` service creates the user `opsiSetupUser` and logs it in.
  - The `explorer.exe` is started as shell.
  - The desktop of the `opsiSetupUser` is visible.
  - After logging in, the upgrade starts after a short delay.
  - The installation files can be retrieved from the server, or stored locally on the client (see `copy_files_locally`).
- `runAsOpsiSetupUser`
  - The `opsiclientd` service creates the user `opsiSetupUser` and logs it in.
- The `powershell.exe` is started as shell and the local `opsi-script` is triggered by it, which executes the upgrade.
  - The desktop of the `opsiSetupUser` is not visible.
  - The installation files must be stored locally on the client.
  - The value set in `copy_files_locally` is ignored and automatically set to true.
- `runOpsiScriptAsOpsiSetupUser` (default)
  - The `opsiclientd` service creates the user `opsiSetupUser` and logs it in.
  - As shell the `powershell.exe` is started and through this the `action_processor_starter.exe` is triggered, which establishes a service connection and then executes the upgrade.

- The desktop of the opsiSetupUser is not visible.
  - The installation files can be retrieved from the server, or stored locally on the client (see `copy_files_locally`).
- If a client is in WAN/VPN mode (automatic detection) this product property is ignored and the installation is performed with the following options: (See also chapter WAN/VPN Mode).
  - The opsiClientd service creates the user opsiSetupUser and logs it in.
  - The powershell.exe is started as shell and through this the local opsi-script is triggered, which executes the upgrade.
  - The desktop of the opsiSetupUser is not visible.
  - The locally cached installation files are used.
- `installfiles_dir`
  - Selection of the folder that contains the installation files.
  - The folder name must always start with `installfiles`!
  - If the product property `automode` should be used, the release ID must be appended to the folder name (e.g. `installfiles2004`)
  - In order to list all `installfiles` directories contained in the package in the product property `installfiles_dir` you may have to reinstall the package via `opsi-package-manager -i` after creating the folders, since the property is filled via the `preinst/postinst` scripts.
- `language_from_language_pack` (See chapter Language Packs)
  - False (default)
    - Select if the client was not installed with an installation media that contains language packs.
  - True
    - Select if the client was installed with an installation media that contains language packs and the client language is provided by a language pack.
- `language_installation_media` (See chapter Language Packs)
  - Specify the language code of the original installation media of the client.
  - Examples:
    - 0407 for German
    - 0409 for english
- `mode`
  - `upgrade` (default)
    - upgrade to the selected release ID.
  - `validate_before_upgrade`
    - Trial run to determine compatibility problems. If no problems are detected the upgrade is started.
  - `validate_only`
    - Trial run to determine compatibility problems.
- `post_oobe`
  - path to `setupcomplete.cmd`
  - More info at: <https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/add-a-custom-script-to-windows-setup?view=windows-11>
- `productkey`
  - License Key to be used for the installation.
- `proxy_ip`
  - Allows to specify the IP of a proxy server.
- `proxy_port`
  - Allows to specify the port number of a proxy server.
- `quiet`

- False
  - The installation starts in interactive mode and requires user input.
  - Useful to get a meaningful error message in case of an error.
- True (Default)
  - The installation starts in silent mode without user interaction.
- `restore_health`
  - False (default)
    - `SFC /scannow` and „`DISM /online /cleanup-image /restorehealth`“ are not executed before the upgrade.
  - True
    - `SFC /scannow`, and „`DISM /online /cleanup-image /restorehealth`“ are executed before upgrade.
- `setup_after_install`
  - Selection of OPSI products to be installed after a successful upgrade.
  - Selectable products (manually upgradeable):
    - `config-win10`
    - `mshotfix`
    - `opsi-client-agent`
- `showoobe`
  - Full
    - Users must click through the Out Of Box Experience (OOBE) dialog when logging in after the upgrade.
  - None (Default)
    - Skips the Out Of Box Experience (OOBE) dialog when a user logs in after the upgrade.
- `telemetry`
  - Disable (Default)
    - No telemetry data of the upgrade will be sent to Microsoft.
  - Enable
    - Telemetry data of the upgrade will be sent to Microsoft.
- `unset_after_install`
  - Selection of OPSI products to be set to not\_installed after a successful upgrade.
- `upgrade_to_version`
  - Selection of the release ID to which the upgrade is to be performed.
  - Will be ignored if `automode` is set to true.
    - Only if no valid release ID could be read via `automode`, the value selected here will be used as fallback.
  - latest (Default)
    - Always refers to the latest release ID that was current at the time of package creation.

## WAN/VPN mode

The package automatically detects if a client is in WAN/VPN mode. The locally cached installation files are used.

## Special features

**Important!** After each change to the installfiles folders, the package must be reinstalled via `opsi-package-manager -i`, so that the .files file is regenerated!

Packages selected by `setup_after_install` cannot be cached until the client is able to successfully contact the server the next time. Mostly this happens after a VPN connection is established after a user logs in.

## Language Packs

Windows installations with installed language packs are a special case. In order to successfully update an installation that uses language packs, a few points must be taken into account.

### Preparation of the installation medium

All required Language Packs must be included in the `install.wim` of the installation medium for the upgrade. It is important that the language packs are downloaded according to the version of the installation medium. For example, Language Packs in version 1909 cannot be included in a 21H1 installation medium. The integration of the Language Packs is done by the tool DISM. Instructions and further information can be found in the following 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](#)

### Expiration

The operating system to be upgraded must be reset to the language of the original installation media prior to the upgrade. For example, if a client was installed from an English installation media containing a German language pack and German was configured as the client's primary language, the language must be set to English before upgrading. After the upgrade, the language can be set back to German if the installation media used during the upgrade has a German Language Pack integrated that matches its ReleaseID.

This process has been automatically included in version 21h2-1 of the Windows 1x upgrade packages. Before the upgrade, the current language is read and saved. Then the language is set to the language of the installation media used to install the client. After the upgrade, the language is set back to the initially read values.

### Client Settings

- The `language_from_language_pack` product property must be set to true.
- The Product Property `language_installation_media` must be set to the four-digit Language Code of the installation media.
  - Examples:
    - 0407 for German
    - 0409 for English

# Upgrade from Windows 11 Previous Versions

The following Windows 11 predecessor versions can be upgraded to Windows 11:

- Windows 10 from version 2004

Before upgrading, all current Windows updates should be applied.

## Debugging

Several circumstances can cause an upgrade to fail:

- Client does not meet the minimum requirements for Windows 11.
- System Reserved Partition too small
- Virus scanner prevents the upgrade
- Incompatible drivers on the client
- Incompatible software on the client
- Wrong installation media
  - Wrong architecture
  - Wrong language
  - Wrong edition
- Activation status of the operating system before the upgrade
- User interaction during the upgrade (e.g. rebooting or switching off the client)

## Evaluation of the exit code

As of version 20.09-1, the exit codes of the upgrade are evaluated in more detail and a meaningful error message with tips on how to fix it is written to the log file. However, the list of exit codes is certainly not complete, since there is no complete listing of all exit codes on the part of Microsoft.

## Included Upgrade Logfiles

The following upgrade logfiles are included in the logfile of the package in case of error (if available) and can be analyzed conveniently in the Configed:

- "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\*APPRAISER\*.xml,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\*CompatData\*.xml,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Rollback\setupact.log,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Rollback\setuperr.log,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Rollback\setupapi\setupapi.dev.log,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\UnattendGC\diagerr.xml,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\UnattendGC\setupact.log,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\UnattendGC\setuperr.log,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\diagerr.xml,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\MigLog.xml,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\ScanResult.xml,,



- "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\setupact.log,,
- "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\setuperr.log,,
- "%SystemRoot%\Panther\*APPRAISER\*.xml,,
- "%SystemRoot%\Panther\*CompatData\*.xml,,
- "%SystemRoot%\Panther\Rollback\setupact.log,,
- "%SystemRoot%\Panther\Rollback\setuperr.log,,
- "%SystemRoot%\Panther\Rollback\setupapi\setupapi.dev.log,,
- "%SystemRoot%\Panther\UnattendGC\diagerr.xml,,
- "%SystemRoot%\Panther\UnattendGC\setupact.log,,
- "%SystemRoot%\Panther\UnattendGC\setuperr.log,,
- "%SystemRoot%\Panther\diagerr.xml,,
- "%SystemRoot%\Panther\MigLog.xml,,
- "%SystemRoot%\Panther\ScanResult.xml,,
- "%SystemRoot%\Panther\setupact.log,,
- "%SystemRoot%\Panther\setuperr.log,,
- "%SystemRoot%\Logs\MoSetup\Bluebox.log,,
- "%SystemRoot%\Logs\SetupDiag\setupdiagresults.xml,,
- "%opsiLogDir%\opsisetupuser.log,,
- "%opsiLogDir%\SetupDiagResults.log,,
- "%opsiLogDir%\sfcdetails.txt,,
- "%SystemRoot%\logs\dism.log,,

## Possible solutions for problems with Windows Upgrade

To find incompatible software and drivers look in the Windows 11 upgrade logfile, or directly in "%SystemDrive%\\$WINDOWS.~BT\Sources\Panther\scanresult.xml,, and %SystemDrive%\\$WINDOWS.~BT\Sources\Panther\Compatdata\*.xml (several individual log files) for the following strings:

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

In case of an incompatible driver you should see the corresponding .inf file (e.g. oem11.inf). The .inf files are located in the folder %SystemDrive%\Windows\INF. Open them with a text editor to find out the driver name. Uninstall the found driver(s) on the client and start the upgrade again.

**Attention!** There are **always** two drivers with the status CompatibilityInfo BlockMigration="True" found (depending on the system in different oemxx.inf files) The associated devices are:

- Microsoft Print to PDF
- Microsoft XPS Document writer

Although these two drivers are displayed with CompatibilityInfo BlockMigration="True", they **do not** prevent the upgrade and can be ignored.

To get a meaningful error message you can start the upgrade with the Product Properties quiet = false and debug = true. **Attention!** This will start the upgrade in interactive mode and require user input! During the upgrade process you will **mostly** get a more detailed error message in the GUI.

The upgrade can be started with the product property mode = validate or mode =

validate\_before\_upgrade. In this case only a test run of the upgrade is started. Any errors that may occur can thus be determined before the actual upgrade.

## **Solutions for problems with the Windows 11 upgrade package**

Unfortunately, problems with the Windows 11 upgrade package itself can occur from time to time. Generally it is recommended to always use the latest version of the Windows 11 upgrade package!

### **Auto Logon problems**

There can be problems with the Auto Logon of the opsiSetupUser. Mostly group policies prevent the temporary opsiSetupUser from logging in. In this case it should be checked if the logon is only allowed for certain local users or if maybe the password policy prevents the login.

Windows dialogs that require user interaction at login may also prevent automatic login. The Windows 11 upgrade package suppresses all known dialogs, but depending on the release ID, new dialogs may be added that prevent logon. In this case a screenshot of the corresponding dialog is always needed, so that a workaround for this can be included in the next version of the Windows 11 upgrade package.