Microsoft

# WINDOWS AND THE SHARED SERVICING MODEL

**Version 1.3, June 2026**

# ABOUT THIS DOCUMENT

Commercial organizations, partners, and original equipment manufacturers (OEMs) may be unaware of the benefits of the shared servicing model and enablement packages (eKB) when moving to newer versions of Windows. As a result, some are not optimizing their update processes. They may continue to test every application and deploy larger feature updates or worse, re-image their devices when it is unnecessary.

In addition, when OEMs end support of hardware models on an earlier version from a shared branch, organizations are required to either purchase new devices or rely on Microsoft to provide a lifecycle extension for their version of Windows.

The goal of this paper is to clarify what a shared servicing branch is, what the benefits are to the organization or user, and outline what the actual impact is of upgrading through an enablement package.

# WHY THIS MATTERS FOR ORGANIZATIONS

For organizations, the shared servicing model is not just a technical detail; it changes the economics and operational risk of staying current. When devices move between Windows versions on the same servicing branch, the update experience is much closer to a monthly quality update than to a traditional operating system replacement. That means faster deployment, shorter maintenance windows, less disruption for users, and less strain on deployment infrastructure. Enablement packages are **dramatically** smaller than full feature updates—for example, a move from Windows 11, version 24H2 to 25H2 can be measured in kilobytes rather than gigabytes.

- **Faster updates with less downtime** because the enablement package acts as a small activation step rather than a full OS swap.
- **Far lower network bandwidth consumption and smaller downloads** for devices already on the same branch, reducing pressure on WAN links, distribution points, and remote sites.
- **Reduced validation effort** because application and hardware compatibility risk is significantly lower across versions that share the same source code, fixes, and regression testing baseline.
- **Less disruption to end users**, since installation is noticeably quicker and more predictable than a traditional feature update.
- **Simpler planning for IT** because testing can focus on newly enabled features instead of re-validating the entire operating system stack.

# INTRODUCTION

Over the years, we have optimized how we develop and service each version of Windows. Some of these optimizations benefit our developers, such as easier ways to add their source code without affecting other teams' work (users indirectly benefit from higher quality). Some benefit organizations and their users, such as reduced package size (by 40%!), combining servicing stack updates (SSUs) with latest cumulative updates (LCUs) for a seamless experience, and faster installations.

A major improvement we have made has tremendous benefits to both Microsoft and you: the shared servicing branch. It enables us to use servicing technology (like that used for the monthly update process) to deliver a feature update via a familiar, fast, and reliable update experience. This process is called feature updates via servicing. Multiple versions of Windows use the same source code branch for development and servicing. This enables Microsoft developers to be more efficient when servicing these versions of Windows. Most importantly, it makes it much easier and quicker for you to keep your Windows devices current and more secure**.** How so? By avoiding application and hardware compatibility issues when moving from one version of Windows to another on the same branch.

That is a bold statement, and it bears repeating. Updating from one version of Windows to another from the same branch greatly reduces the risk of application or hardware compatibility issues**.** Each version on a shared servicing branch shares the following:

- The source code
- Security and non-security fixes
- Compatibility and regression testing done prior to release

The only difference between versions is the set of features "turned on" in the newer version.

So far, there are two shared source code branches for Windows 11 releases: the 22H2/23H2 branch and the branch for 24H2/25H2/26H2.  Its first release, 21H2, was not on a shared branch.

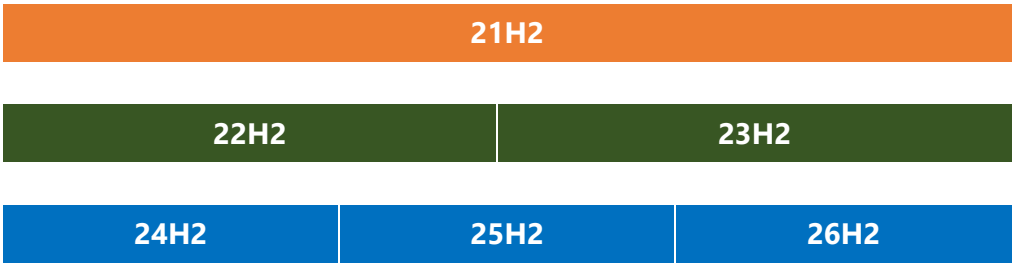


*Figure 1. Graphical representation of the source code branches for Windows 11*

What does this mean for moving from Windows 11, version 25H2 to 26H2? We introduced new features without regressions or compatibility issues because 26H2 is the same source code with only the additional features turned on. If an issue affects 26H2, it will also affect 25H2, and vice versa. Likewise, an update that would fix that issue is the same update for both versions. It is the

same source code, the same update package to install. This is not to say you shouldn't test; but those tests should focus on the new features, not a full complement of application and device compatibility tests or certifications.

# WHAT IS A BRANCH?

Simply stated, a branch is a repository of source code used to build the Windows operating system. Developers create their code. When ready, they check the code into their team's "local" branch. Ultimately, it's rolled into the "main" branch of all the source code that makes up a specific version of Windows.

When a version of Windows is ready to be released, the final code within the main branch is built, packaged, and shipped through various media types. This includes ISO images and feature update packages available on Windows Update.

All versions also require regular servicing or updating with security or quality fixes. When that version needs to be serviced, that code is replaced with the newer code. It's then tested and released on Windows Update via an LCU package on Patch Tuesday, the second Tuesday of each month.

## Branches and Windows versions

Historically, every version of Windows came from its own branch. The previous version's branch would "fork." That is, it would be copied to a new branch as a baseline for the new version of Windows and all its new and changed features to be coded from. A branch can have multiple products/editions, each having the same final build number. A branch remains in service until the final product shipping from it reaches the end of its lifecycle. After that time, the branch is shut down, and no more servicing occurs.

Here is a list of some versions of Windows, consolidated by branch:

| Windows release(s) | Build number |
|---|---|
| • Windows 10, version 1809 | 17763 |
| • Windows 10, version 1903<br>• Windows 10, version 1909 | 18362<br>18363 |
| • Windows 10, version 20H1<br>• Windows 10, version 20H2<br>• Windows 10, version 21H1<br>• Windows 10, version 21H2<br>• Windows 10, version 22H2 | 19041<br>19042<br>19043<br>19044<br>19045 |
| • Windows 11, version 21H2 | 22000 |
| • Windows 11, version 22H2 | 22621 |

| Windows release(s) | Build number |
|---|---|
| • Windows 11, version 23H2 | 22631 |
| • Windows 11, version 24H2<br>• Windows 11, version 25H2<br>• Windows 11, version 26H2 | 26100<br>26200<br>26300 |

*Table 1. Windows branches (not a complete list)*

When a new version of Windows is released, devices are typically updated through what we call an "OS swap." That's when the entire operating system is replaced with the updated code. This is the reason that the package and download sizes, as well as the overall update time a user experiences, can be significant.

## Servicing Windows branches

Once a version of Windows has been publicly released (for example, Windows 10, version 1903), it continues to be serviced via monthly updates. These may include security and non-security fixes. These updates are typically released on the second Tuesday of each month (Patch Tuesday) but may also be released "out-of-band" in case there is a critical security fix required.

These updates – which are released through the LCU – are branch-specific. If there's a security fix, for example, that is required on all versions of Windows 10, the Windows developer must create a fix appropriate for each of the branches that Windows 10 releases come from. Each version has its own LCU that contains the appropriate fix.

For a commercial organization, the more versions of Windows 10 in the environment, the more version LCUs needed to be deployed each month.

# THE SHARED SERVICING BRANCH

With Windows 10, version 1909, we did something completely different. Instead of forking the Windows 10, version 1903 branch to create a new one for 1909, *we used the same branch* and simply added the new or enhanced features to the 1903 branch. When the new code was ready, we included it in the monthly updates (LCUs). Those were released as the monthly updates for Windows 10, version 1903.

That new code, however, was disabled. Those new features were not active, and version 1903 remained the same. Think of it as having the new feature code for what would become Windows 10, version 1909 being slowly staged on the 1903 devices. It remained dormant until all the new code was on the device, ready to be enabled at the right time. That's when we officially released Windows 10, version 1909.

We are using a shared servicing branch for Windows 11, versions 24H2, 25H2 and 26H2. Since this branch also includes the Long Term Servicing Channel (LTSC) editions of Windows 11, it is a separate branch from Windows 11, version 23H2.

The following diagrams explain how monthly updates (LCUs) are cumulative to all versions of Windows released from the same servicing branch. The payload (i.e., new features) for the next version are included in the LCU in a disabled state. Note: The diagram is not meant to display the overall size of each update.

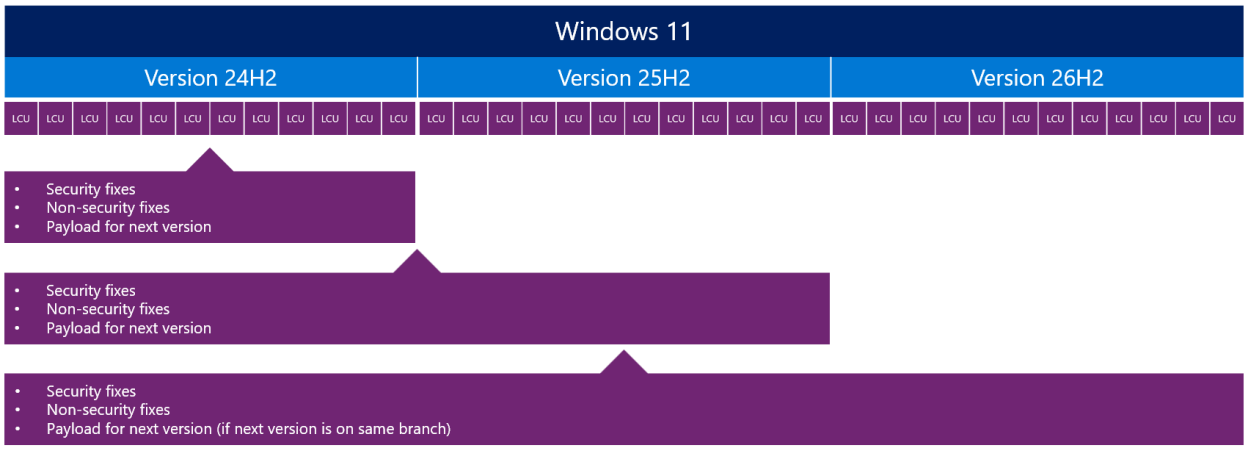


*Figure 2. Shared servicing of Windows 11, version 24H2/25H2/26H2 includes shared security fixes, non-security fixes, and payload for next version*

## Different experience between feature updates and LCUs

| | Feature updates | Latest cumulative updates (LCUs) |
|---|---|---|
| Frequency | Delivered once a year. | Delivered monthly. |
| Replacements | Replace the entire OS. | Replace only the binaries (components) that are serviced. |
| Windows 11 servicing | Deliver major OS releases. | Deliver monthly changes on top of the last major OS release. |
| Numbering changes | Change the build version number. | Change the revision number. |
| Installation | Very noticeable to users (involve the "flashing colors" screen).<br><br>As image-based updates, installation involves laying down an entire pre-staged OS image onto a system. It gathers settings and data from your existing OS, specializes the OS image to the | Less intrusive to users, requiring a quick restart. |

| | Feature updates | Latest cumulative updates (LCUs) |
|---|---|---|
| | machine, and then applies the data and settings onto the image.<br><br>Can run compatibility checks, inform the user, and/or take other action before the start of installation. Action can include informative messages to resolve issues, as well as blocks that can be messaged to the user. | |
| Delivered changes | Can deliver any changes that have been shipped as part of major Windows releases. | Deliver more limited changes, consisting of improving security and functionality. |

*Table 2. Contrasting feature updates and LCUs with respect to frequency, replacements, servicing, numbering, installation, and delivered changes.*

## Upgrading to the next version of Windows on the same servicing branch

When the next version of Windows is completed, **new media (ISO)** is created, and both a **Feature Update** and an **enablement package (eKB)** are released to Windows Update. The ISO is a complete OS installation; it and the Feature Update enables the "OS swap" from Windows versions earlier than 24H2, versions that would not receive the 25H2 monthly updates containing the disabled code.

It is the **eKB** that enables a device currently running Windows 11, version 25H2 to quickly become Windows 11, version 26H2:

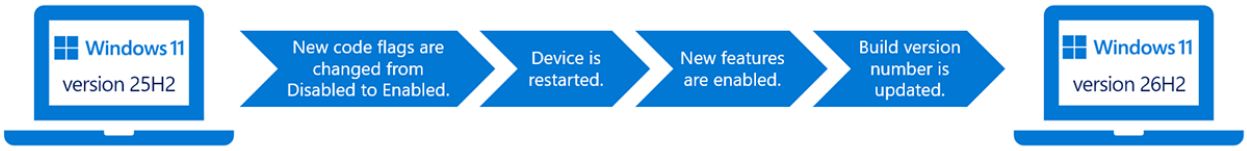


*Figure 3. The four stages of an update from one version to another: New code flags change from Disabled to Enabled, device is rebooted, new features are enabled, and the Windows build version number is updated.*

# CASE STUDY: MOVING FROM WINDOWS 11, VERSION 23H2 AND VERSION 24H2 TO VERSION 25H2

The complete feature update from Windows 11, version 23H2 (which is NOT on the same servicing branch as Windows 11, version 25H2) includes the following components and sizes (x64, US English, Business editions example):

| Component | Description | Size |
|---|---|---|
| Base 24H2 image | Base image of Windows 11, version 24H2 | 5.3 GB |
| Final pre-25H2 LCU | Contains the complete payload for additions for 25H2 | 887 MB |

*Table 3. Components of a feature update for Windows 11, version 25H2*

This feature update, which would be required to upgrade from Windows 10, version 23H2, for example, is **close to 6.5GB!**

However, if the device was up to date and running Windows 11, version 24H2, you'd only download the enablement package, which is just **174 KB**, about three-thousandths of one percent of the roughly 6.5 GB full feature update.

**174 KB VS. 6.5 GB:**
**THE ENABLEMENT PACKAGE IS ABOUT THREE-THOUSANDTHS OF ONE PERCENT THE SIZE OF THE FULL FEATURE UPDATE**

# WHEN IS AN ENABLEMENT PACKAGE POSSIBLE?

Not all versions of Windows can share the same branch. Here are some points we consider when determining if a branch can be shared with the next version of Windows:

- Are both Windows client and Server editions being developed from the branch?
- Are there any discrepancies between short-term and long-term lifecycle editions?
- Are there specific new features or architecture changes that cannot be disabled by default?

Because of this, not all updates can be as simple as deploying the eKB. For example:

☒ **Windows 11, version 21H2** is not from the same branch as **Windows 10**; hence, a full "OS swap" is required.

☒ **Windows 11, version 22H2** is not from the same branch as **Windows 11, version 21H2**; hence, a full "OS swap" is required.

☑ **Windows 11, versions 22H2** and **23H2** do share the same servicing branch, so upgrading from 22H2 to 23H2 through an enablement package is possible.

☒ **Windows 11, version 24H2** is not from the same branch as **Windows 11, version 23H2**; hence, a full "OS swap" is required.

☑ **Windows 11, versions 24H2, 25H2** and **26H2** are on the same servicing branch, so upgrading from **Windows 11, versions 24H2** or **25H2** to **Windows 11, version 26H2** through an enablement package is possible.

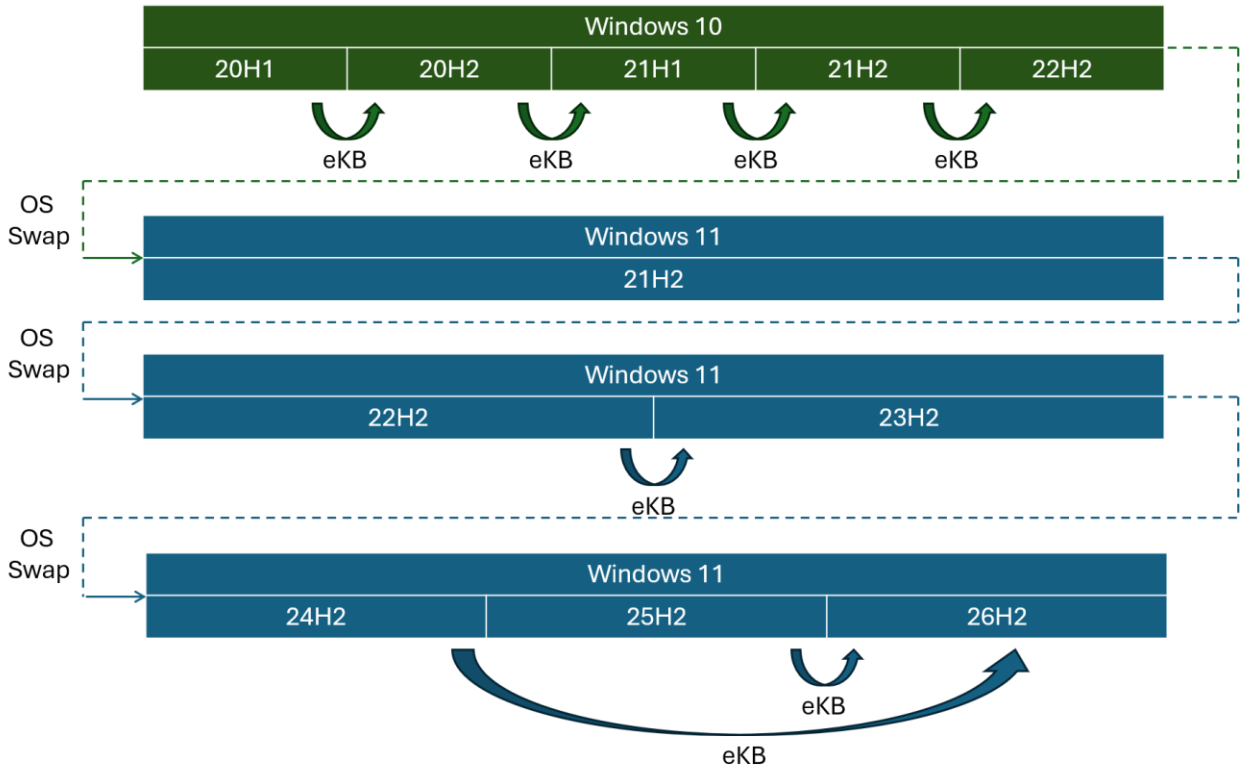


*Figure 4. How to update from one version of Windows to another depends on their branches.*

We try to optimize a branch to be shared across versions whenever possible and will broadly and clearly communicate when a shared servicing branch is used. This information and guidance should assist your planning for future Windows updates.

# CONCLUSION

In conclusion, understanding which Windows versions share a servicing branch is not just useful for technical planning; it has direct business value. Shared servicing branches and enablement packages can help organizations move faster, use far less network bandwidth, reduce download size, minimize user disruption, and lower the amount of compatibility validation required between versions on the same branch. In practical terms, that means a more predictable update motion, lower operational cost, and a simpler path to keeping devices current and secure. Microsoft will continue to communicate when shared servicing branches are available so organizations can use this model to plan more efficient Windows updates.