

Windows and the shared servicing model

Version 1.1, February 2024

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.

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 provides 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 all 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 have been two shared source code branches for Windows 10 releases: the 1903/1909 branch and the branch for 20H1/20H2/21H1/21H2/22H2.

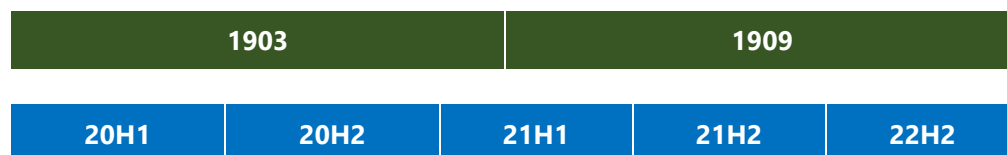


Figure 1. Graphical representation of two source code branches for Windows 10

What does this mean for moving from version 20H1 to 22H2? We introduced a few new features without regressions or compatibility issues because 22H2 is the **same source code with only the additional features turned on**. If an issue affects 22H2, it will also affect 20H1, 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 the 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
<ul style="list-style-type: none">Windows 7 SP1Windows Server 2008 R2	7600
<ul style="list-style-type: none">Windows 8Windows Server 2012	9200
<ul style="list-style-type: none">Windows 8.1Windows Server 2012 R2	9600
<ul style="list-style-type: none">Windows 10, version 1507	10240
<ul style="list-style-type: none">Windows 10, version 1511	10586
<ul style="list-style-type: none">Windows 10, version 1607Windows Server 2016	14393
<ul style="list-style-type: none">Windows 10, version 1809Windows Server 2019	17763

Table 1. Windows branches (not a complete list)

As mentioned above, when a new version of Windows is released, devices are typically upgraded 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 upgrade 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.

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.

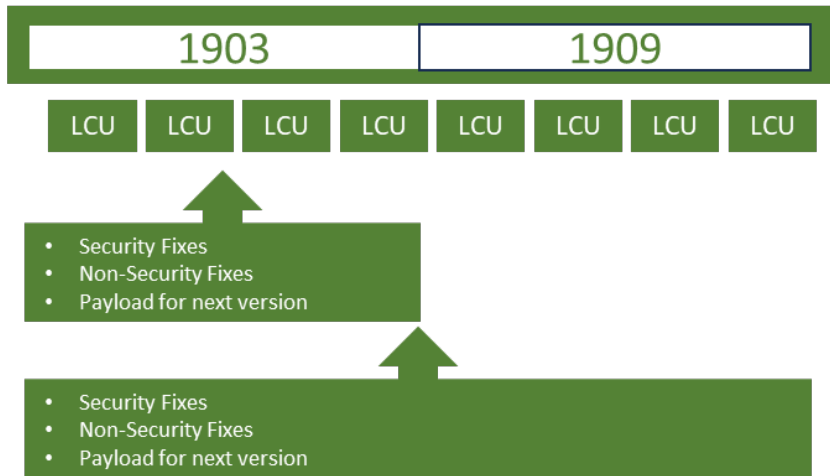


Figure 2. Shared servicing of Windows 10, version 1903/1909 includes shared security fixes, non-security fixes, and payload for next version

We also used a shared servicing branch for Windows 10, versions 20H1 – 22H2, though this was a *different* branch than the one used for 1903/1909.

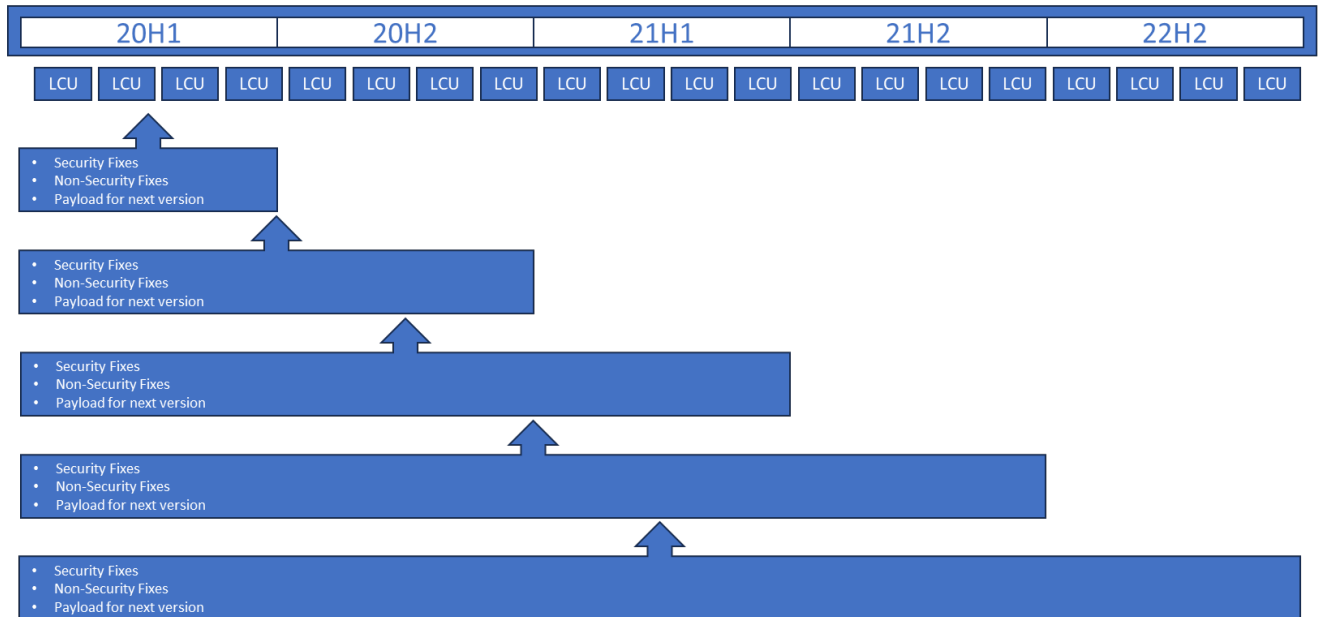


Figure 3. Shared servicing of Windows 10, version 20H1 through 22H2 includes shared security fixes, non-security fixes, and payload for next version

So far, we have been able to utilize shared servicing branches twice: for the 1903/1909 branch and the 20H2 – 22H2 branch. With the release of Windows 11, version 23H2, there are three shared servicing branches:

Windows release(s)	Build number
--------------------	--------------

<ul style="list-style-type: none"> Windows 10, version 1903 Windows 10, version 1909 	<ul style="list-style-type: none"> 18362 18363
<ul style="list-style-type: none"> Windows 10, version 20H1 Windows 10, version 20H2 Windows 10, version 21H1 Windows 10, version 21H2 Windows 10, version 22H2 	<ul style="list-style-type: none"> 19041 19042 19043 19044 19045
<ul style="list-style-type: none"> Windows 11, version 22H2 Windows 11, version 23H2 	<ul style="list-style-type: none"> 22621 22631

Table 2. - Versions of Windows on shared servicing branches

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 10 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	<p>Very noticeable to users (involve the “flashing colors” screen).</p> <p>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 machine, and then applies the data and settings onto the image.</p> <p>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.</p>	<p>Less intrusive to users, requiring a quick restart.</p>

Delivered changes	Can deliver any changes that have been shipped as part of major Windows releases.	Deliver more limited changes, consisting in improving security and functionality.
--------------------------	---	---

Table 3. 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

At the time we complete the next version of Windows, new media is created: the ISOs and a feature enablement package, called eKB. The ISOs are full “feature updates” that enable the OS swap from Windows versions earlier than 22H2. These are the versions that would not receive the 22H2 monthly updates containing the disabled code.

It is the eKB that enables a device currently running Windows 10, version 21H2 to quickly become a Windows 10, version 22H2:



Figure 4. The four stages of an upgrade from one version to another: New code flags change from Disabled to Enabled, device is rebooted, new features are enabled, and build version number is updated.

Example: What comprises Windows 10, version 22H2

The complete feature update to upgrade a previous version of Windows 10 to version 22H2 includes the following components and sizes (x64, US English, Business editions example):

Component	Description	Size
Base 20H1 image	Base image of Windows 10, version 20H1	3 GB
Final non-22H2 LCU	Contains the complete payload for additions 20H1-22H2	715 MB
22H2 eKB	Enables new 22H2 features and changes version number	6 KB
Security Stack Update	Enables the installation of the eKB	14 MB

Table 4. Components of a feature update for Windows 10, version 22H2

This feature update, which would be required to upgrade from Windows 10, version 1909, for example, is **close to 4GB!** It would still be a good idea, since version 1909 is out of support.

However, if the device was up to date and running Windows 10, version 21H2, you’d only download a **6KB enablement package!** That means that you’d have the latest monthly update (LCU) installed, which can include the SSU. You’d only need to deploy the feature enablement package (eKB) for the device to upgrade to Windows 10, version 22H2!

Conclusion

Using a single **shared servicing branch**, Microsoft can develop and service more consistently higher quality versions of Windows, while enabling devices to be updated to newer versions very easily and quickly. Commercial organizations can easily upgrade from one version to the next without worrying about quality regressions or compatibility issues with applications or hardware. An upgrade from one version to another on the same branch does not require additional testing, and there is no burden on the corporate network infrastructure or end user experience. An upgrade literally takes just a few minutes.

Since each version of Windows on a shared servicing branch contains the same source code, applications and hardware that work on the earlier version ***should work exactly as expected*** on the newer version without modification.

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 upgrades can be as simple as deploying the eKB. For example, Windows 11, version 21H2 is not from the same branch as Windows 10, version 22H2. Hence, a full "OS swap" is required. Windows 11, version 22H2 is not from the same branch as Windows 11, version 21H2 either, so an eKB is not available. (Though, as previously mentioned, Windows 11, version 22H2 and 23H2 do share the same servicing branch, so upgrading from 22H2 to 23H2 will be a snap!)

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 upgrades.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. Microsoft makes no warranties, express or implied, in this document.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2023 Microsoft Corporation. All rights reserved.

Microsoft, list Microsoft trademarks used in your white paper alphabetically are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.