WHITE PAPER–SEPTEMBER 2022

Modern Management Essentials: OS Lifecycle Management

vmware®

Table of Contents

vmware®

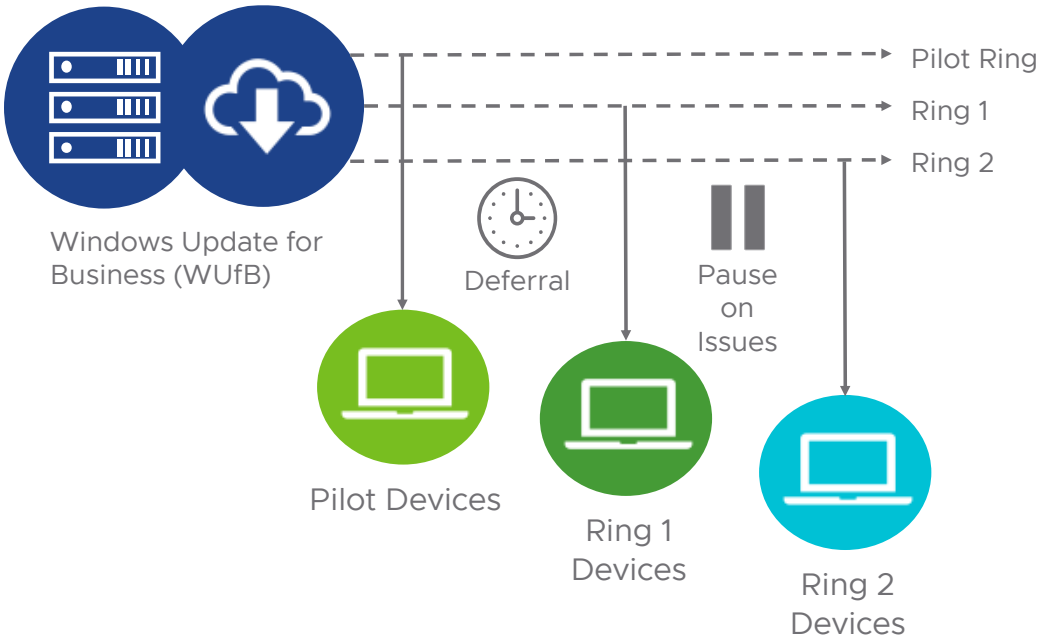
Overview

This whitepaper provides an overview of the approach to Windows Modern Management OS Lifecycle Management, focusing on rapid deployment of the ideal configuration rather than niche configurations. It also describes the key design decisions to assist in adjusting the configuration to suit the more complex environments.

A common legacy OS Lifecycle Management methodology used a "green button" approach. This approach controlled which updates are visible and when, by manually approving updates for specific logical groups of devices on a scheduled basis. The process was initiated by a manual approval for each logical group of devices, with the rationale of deploying in stages, reducing risk, by providing control in case of issues. When an update or set of updates successfully deploys to the first logical group of devices, an Administrator would then press the "green button" to deploy to the next logical group of devices, and so on.

This methodology is time consuming for Administrators as it is largely a manual process, relying on the Administrator to select and approve updates that are applicable to their fleet. Researching each update is also time consuming, and not approving individual updates could lead to vulnerabilities remaining exposed. And lastly, the lag between the release of an update and Administrator approval, increases the time to achieve a compliant state on devices.

The Windows Modern Management OS Lifecycle Management approach is to move to a "red button" approach where updates are automatically approved by policy; devices are constantly checking on availability of updates that are approved; installing those updates within the maintenance window; and only pausing installation of updates if an issue is found (the "red button").



In the “red button” approach, each logical group of devices has an increasing delay before updates are available to those devices. This approach provides time for an Administrator to pause updates in case an issue arises in order to find a resolution or wait for Microsoft to release a replacement update.

Policy automates the approval of all updates, as well as provides control to include or exclude driver updates, ensuring updates that Microsoft recommends are deployed in a timely manner. Combined with the automated delay or deferral for each logical group of devices, updates get deployed and installed within the configured timeframe set to match the organization’s security and risk policy. And this process is not reliant on an Administrator to start the process each day, week, or month, enabling faster time to compliance for OS Updates.

The Windows Modern Management OS Lifecycle Management approach provides efficiencies by reducing administration, reduces the time to get updates deployed to devices, whilst providing the means to deploy updates to a subset of devices in order to test them.

Purpose of This Tutorial

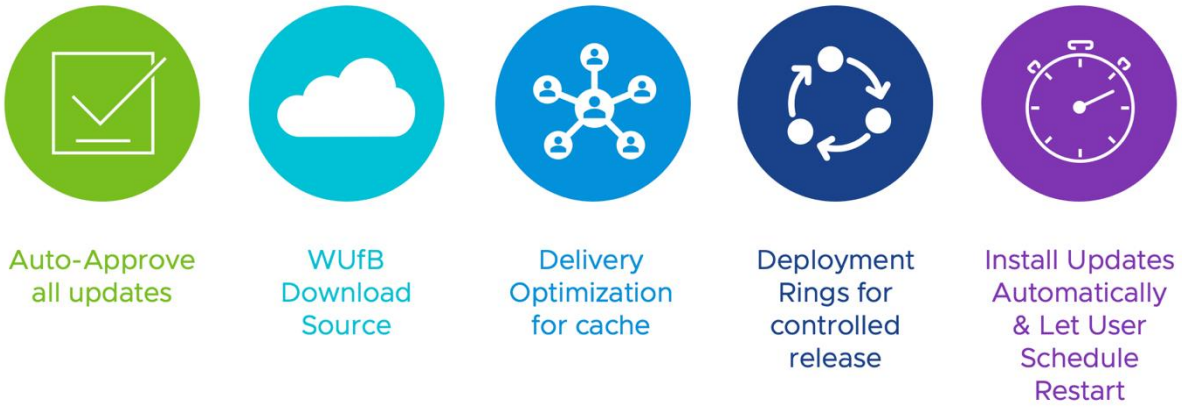
VMware Workspace ONE® enables cloud-native modern management to automate IT operations, harden security, and deliver ready-to-work experiences across every Windows device—whether on or off the enterprise network. This document walks through the design considerations and best practices for Windows lifecycle management.

Audience

This guide is for PC life cycle management (PCLM) administrators and Workspace ONE IT administrators. Familiarity with Windows 10 and above is assumed, including Active Directory, or Azure Active Directory. Knowledge of additional technologies such as VMware Workspace ONE® Access (formerly VMware Identity Manager) and VMware Workspace ONE® UEM and Microsoft Endpoint Configuration Manager (formerly System Center Configuration Manager or SCCM) is useful.

Planning Lifecycle Management for Windows

This chapter helps you plan the lifecycle management process for your Windows device fleet. The recommended approach for Windows lifecycle management is to auto-approve all updates, use Windows Update for Business (WUfB) as the download source, cache updates and manage downloads with Delivery Optimization, use deployment rings to control deployment, and to install updates automatically during a daily maintenance window.



While the above configuration fits with most customer requirements and how Workspace ONE is designed to support Windows Updates, it may be necessary to deviate from the ideal configuration. To determine what deviations are required, you must ask yourself some questions to determine what you really need. This section walks through the primary design decisions you must consider.

Design Decision: Auto-Approve or Manual Approval

Service Channels determine how stable updates are before they are made available to devices, how often Feature Updates are released and how long a release will be supported by Microsoft.

Deploying Quality Updates (individual and cumulative) as well as Feature Updates (upgrades) to your fleet in a controlled way allows you to validate whether your applications have any compatibility issues with those updates and mitigate if there are issues. Standard, off-the-shelf applications shouldn't have issues. However, the Windows ecosystem is huge, and some software may have hard requirements to specific versions of Windows or Windows components.

It is important to use the correct Service Channels for your devices. Most devices should subscribe to the Semi-annual Channel in order to receive quality and feature updates with the ability to defer those updates from being delivered to devices. However, Windows 10 Enterprise LTSB Edition has Long-Term Servicing Channel built in and cannot be changed. Devices with this Windows Edition installed, will get Quality Updates for up to 10 years, but will not get any Feature Updates.

Windows Insider Program for Business	> only UAT devices Feature Updates - **Continuous** Quality Updates – **Continuous** Support – **NA** Editions – **All** Deferral – **NA**
Semi-annual Channel	> the majority of devices Feature Updates – **Twice a year** Quality Updates – **Continuous with Monthly cumulative** Support – **18 months** Editions – **All** Deferral – **Up to 30 days Quality + 365 days Feature Updates**
Long-term Service	> only certain use cases Feature Updates – **None** Quality Updates – **Continuous with Monthly cumulative** Support – **5 years standard + 5 years extended** Editions – **Enterprise LTSB** Deferral – **up to 30 days Quality Updates**

Ref: https://docs.microsoft.com/en-us/windows/deployment/update/waas-overview#servicing-channels

You can use deployment rings to automate the approval and staggered deployment of updates to logical groups of devices. Workspace ONE UEM provides logical grouping of devices with Smart Groups and policy settings by deploying Profiles targeted at those Smart Groups. These Smart Groups are akin to deployment rings.

Consider using deployment rings if you currently:

- Auto-approve some categories of updates
- Approve cumulative updates to test devices on a programmed cycle (for example patch Tuesday)
- Approve updates to test them without knowing what each fix is

Deployment rings with auto-approved updates and deferral days configured, provide a mechanism to automate the above business process whilst continuing to provide control. Note that Microsoft will still maintain the ability to deploy emergency patches bypassing the approval flow, allowing Updates to install irrespective of the category approved.

In contrast, if you approve updates individually in order to maintain strict control of software versions, then you may need to continue to use a manual approval process. In this scenario, you should deploy Windows Server Update Service (WSUS), create and maintain Device Groups and manually approve the appropriate updates for each Device Group. In addition, a single Workspace ONE Windows Update Profile would be deployed to devices configuring the WSUS server the device should report to.

Important: Either auto-approve everything using Deployment Rings and Workspace ONE Profiles OR manual approve everything with WSUS Device Groups.

The majority of quality updates are categorised in the Critical, Definition, and Security update categories. Additionally, Cumulative updates are also categorised as Security updates. If you auto-approve those specific

categories of updates already, you should auto-approve all quality updates, and use a deferral with a deployment ring methodology to control the release of those updates.

Important: Due to the use of a dated CSP (./Vendor/MSFT/Update), Microsoft does not recommend approval based on category. Further, reliability issues with this CSP cause some updates to categorise incorrectly and exclude some dependencies - making the update a blocker. As such, DO NOT enable the "Require Update Approval" and "Auto-Approved Updates" controls within the Workspace ONE OS Updates Profile or utilise the "Update/ApprovedUpdates/Approved Update Guid" CSP. Either auto-approve everything, or manually approve everything.

Design Decision: Target Release Version or Deferral for Feature Updates?

Feature Updates (upgrades) are released on a different cycle and require different test, validation and release controls to ensure applications do not have any compatibility issues. As such, Feature Updates need to be managed with a different policy aligned to a specific Feature Update Deployment Ring as distinct from the Quality Update Deployment Ring.

To set the test, validation and release cycle for feature updates you configure either of the following policy settings:

- Target Release Version – Use if business applications require testing / certification by the vendor, or the OS Update/Lifecycle policy states N-1. TargetReleaseVersion can be used to lock devices into a feature update version and continue with quality updates until end of service.
- Product Version – Use if business applications require testing / certification by the vendor for a specific Windows OS version such as Windows 10 or Windows 11. ProductVersion can be used to force the device to upgrade or stay on the specified version.
- Deferral – Use if the testing cycle of feature updates lags behind feature update releases by a number of days, weeks, or months, AND the same Deployment rings are used for Quality and Feature Updates. Set Defer Feature Updates (Days) to 0 (zero) if using Target Release Version. Keep in mind, you can only defer a feature update for 365 days. After deferral, the latest feature update is provided, not the next feature update. For example, if the latest feature update released for Windows 10 is 2004 and if you defer for 365 days on 1903, the device will get the 2004 feature update and miss 1909.

Design Decision: WUfB or WSUS?

In terms of download source, WUfB is the ideal choice. WUfB supports device mobility and integrates with delivery optimization to overcome network topology and network constraints that relate to downloading of updates.

However, you may need to utilise a WSUS server or multiple strategically placed WSUS servers to cache updates. The network topology, device numbers in each site, and mobility of devices determine if WSUS servers are required and where WSUS servers are required. Consider using WSUS as your download source if:

- The majority of your devices are non-mobile desktops
- You have a single internet gateway for your entire organisation
- Your network topology is a hub and spoke with limited WAN connectivity
- You must connect to the internet via proxy (cloud or VPN) to meet security requirements

You can simplify profile assignment to devices in WSUS sites using sensors and Workspace ONE Intelligence automation. Refer to the Dynamic Assignment section below. While WSUS is a valid option for some environments as described above, WSUS has the following cons:

- It is designed for on-premises, isolated or non-internet connected deployments, and therefore requires server infrastructure.
- It is inefficient at updating devices outside the corporate network.

Design Decision: One or multiple Delivery Optimization groups?

Delivery Optimization provides several benefits that all help minimise downloads across links for sites with multiple devices and should be on by default. Benefits include peer to peer downloads, bandwidth management, download recovery and chunking, and disk caching.

However, your network setup determines how many delivery optimization groups you require:

- Single Delivery Optimization Group – If you have a simple distributed network or your devices are always mobile, then a single delivery optimization group with the download mode set to **Use Peers on Same Local Network** is the best approach.
- Multiple Delivery Optimization Groups – If you have a large, multi-site, distributed network with MAN/WAN links, then consider separate delivery optimization groups for each site. This means each site has its own delivery optimization profile with a unique delivery optimization GroupId.

You can simplify profile assignment to devices in separate sites using sensors and Workspace ONE Intelligence automation. Refer to the Dynamic Assignment section below.

Design Decision: Deployment Rings

Deployment rings provide control of when updates are released to logical groups of devices and have been used and recommended by Microsoft for many years. Smart Groups are Workspace ONE's method to target these logical groups of devices.

Each deployment ring has a different policy assigned to a Smart Group. The primary difference between deployment ring policies are their deferral policies, which determine when updates are approved and visible to devices. These deferrals provide time to identify and resolve update issues. It allows a staggered approach to deployment across Line of Business (LOBs) and sites to manage the risk to the business if there is an issue, as well as reducing load on the service desk. To be clear, all Quality Updates are auto-approved. Control is provided by staggering deployment. If there is a problem, Pause Updates if needed.

Create deployment rings to match the test, validation and release cadence needed. Consider spreading

devices from sites across deployment rings to reduce the risk of an issue effecting the entire site as well as create a cache of updates for devices in subsequent deployment rings.

If Feature Updates (upgrades) are released on a different cycle and require different test, validation and release controls in your organization, create separate Feature Update deployment rings. Each Feature Update deployment ring will have a corresponding Feature Update Profile.

Design Decision: Install Updates Automatically & Let User Schedule Restart?

When to install updates and restart a device are critical user experience and device compliance controls, and therefore critical design decisions.

The default policy on Windows 10 and above is Auto install and restart. Updates are downloaded automatically on non-metered networks and installed during the "Automatic Maintenance" window when the device isn't in use and isn't running on battery power. If automatic maintenance is unable to install updates for two days, Windows Update will install updates immediately. If a restart is required, then the device is automatically restarted when the device isn't actively being used. Devices are updated quickly, but it increases the risk of accidental data loss caused by an application that doesn't shut down properly on restart. For more information, see Automatic Maintenance article.

- Auto-Restart Notification - Windows Update toast notifications disappear after 15 minutes. This can cause issues such as unexpected restarts. Consider setting this policy to User Dismissal (2) in order to keep the toast notification present until the user acknowledges it.

Alternate policies that provide greater control to the user, and a better user experience are:

- Auto install the update and then notify the user to schedule a device restart
- Auto install and restart at a specified time

Auto install the update and then notify the user to schedule a device restart

With this policy, update download and install behaviour is the same as the default described above. However, if the installation requires a restart, the end user is prompted to schedule the restart. The end user has up to seven days by default to schedule the restart and after that, a restart of the device is forced.

This is the recommended policy as enabling the end user to control the restart time reduces the risk of accidental data loss caused by applications that don't shut down properly on restart.

Additional policy settings can be used to control the maximum number of days the device tries to install updates within the “Automatic Maintenance” window and the maximum number of days the user has to schedule the restart. These policy settings combined reduce the ‘time to compliance’ for the device.

- Configured Deadline for Quality Updates - Allows admins to specify the number of days before quality updates begin downloading and are installed on a device automatically. This policy forces the install of updates when the deadline in days is reached if they were not able to be installed during the “Automatic Maintenance” window.

- Auto Restart Deadline – If you have a compliance requirement for devices to update within a set number of days, then balance deferrals with the automatic restart deadline. Alternatively, consider the following:
    - Auto Restart Deadline Grace Period - Allows the admin to specify a minimum number of days until restarts occur automatically for quality updates. Setting the grace period might extend the effective deadline set by the deadline policy. Do not configure unless this fits with the desired user experience.
    - Engaged Restart Deadline or Snooze – Delays device restart by allowing users to snooze or reschedule the restart. This extends the time a device remains non-compliant. Do not configure unless this fits with the desired user experience.

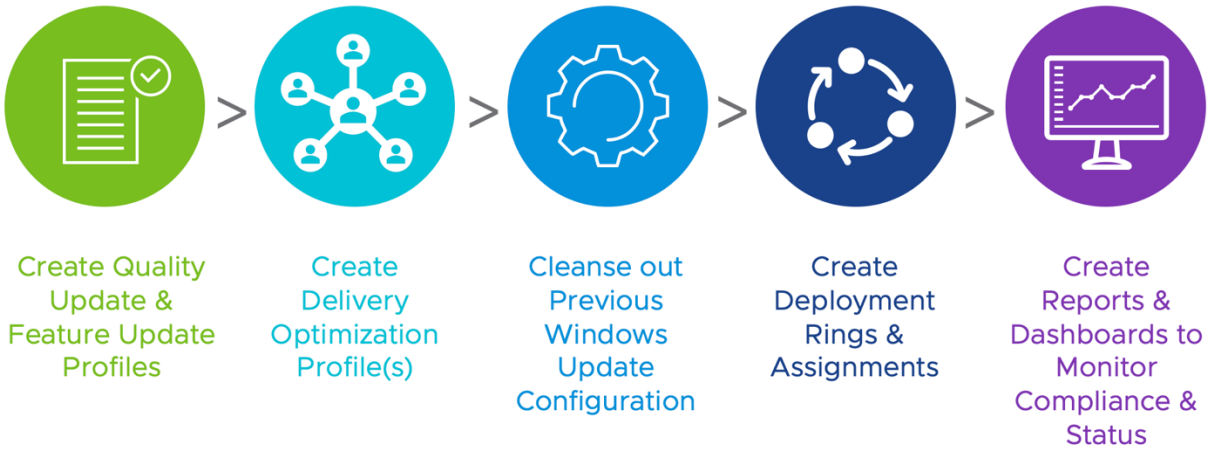
Auto install and restart at a specified time

For use cases that require a specific maintenance window, where installation of updates and device restarts occur regardless of user activity, then this policy should be used. Example use cases include retail, call centre and kiosk devices that.

With this policy set, updates are downloaded automatically on non-metered networks, however installation occurs at the day and time specified in the policy. Automatic installation happens at this time and device restart happens after a 15-minute countdown. If a user is signed in when Windows is ready to restart, the user can interrupt the 15-minute countdown to delay the restart.

Configure Lifecycle Management for Windows

The lifecycle management process can be built in five phases shown in the diagram below. This section walks you through those phases and provides configuration guidance and best practices.



Create quality update, feature update, and delivery optimization profiles

As described above, quality update and feature update policy should align with the test, validation, and release cadence of that update type for the organization. If different, then separate policies should be created and deployed as separate profiles and targeted to separate Smart Groups for each quality and feature update deployment ring.

	Quality Updates	Feature Updates
Deployment Cadence	Faster cadence	Slower cadence
Deployment Rings	Deploy in large groups of devices, increasing the number of devices in each subsequent deployment ring.	Deploy in smaller groups of devices, with similar numbers of devices within a larger number of groups.

If multiple delivery optimization groups are required, the delivery optimization policy should be separated from Quality or Feature Update policy and deployed as separate profiles, targeted to separate Smart Groups. In addition, because delivery optimization configuration is largely based upon network topology, having delivery optimization policies deployed to devices in a separate profile provides more flexibility and reduces the total number of profiles created, maintained, and assigned.

Configure a quality update profile

To set up the ideal configuration, consider the following policies. A table is provided near the end of this document with links to the CSP documentation of each setting for further information.

- Service Channel – Set to Semi-Annual (16).
- Install Updates Automatically but Let User Schedule Restart – Set to 1, this option to address most use cases.
  Alternatively, if a specific maintenance window is required consider the following settings:
    - Install Updates Automatically & Restart at Specified Time – Set to 3 this is a good option for environments or use cases with minimal user interaction such as kiosks, contact centres and process worker terminals, that also require installations and restarts within a specified maintenance window.
    - Schedule Install Day – Set the day of the maintenance window
    - Schedule Install Time – Set the time to start the install, within the maintenance window
- Use Intelligent Scheduling – Install updates outside of active hours and restart at 2am in daily maintenance period.

  It is important to remember that Windows Updates will install and restart out of 'Active Hours' if the device is not actively being used. If it is active, the device will retry the install operation for 2 days and retry a restart for 7 days. If not able to install, then it will automatically install or restart the following day irrespective of 'Active Hours'. It is also important to remember that some critical updates will install without waiting the deferral time and restart a device automatically during 'Active Hours'.
    - Active Hours Maximum (Hours) – Set to total hours. This should match the difference between the start and end times.
    - Active Hours (Start) - Set to start time for example set to 8 will start Active Hours at 8am.
    - Active Hours (End) – Set to end time for example set to 20 will end Active Hours at 8pm.
    - Wakeup OS for Daily Scheduled Maintenance – Set to 1. Configure this policy to wake a device and allow it to install updates and restart. By default, devices sleeping or hibernating during the daily scheduled maintenance window do not install or restart from updates. Instead, the install and restart may occur during active hours after Windows Updates tries for 2 days.
- Require Update Approval – Set to 0.
  Alternatively, if you approve updates individually, then you need to stay with manual approval process. This requires you to manually approve those individual updates within WSUS Server Device Groups. In this situation, a single Workspace ONE Profile is required to deploy to devices with the following settings:
    - Require Update Approval – Set to 1.

Note: It is not recommended to manually approve updates within Workspace ONE. Utilise WSUS Server to manage device groups and approve updates to those device groups.

- Update Service URL – **WSUS Server URL**
  Note: If Auto-Approving Updates and you want to use WUfB as your download source ***Do Not Configure*** this setting.

- **Auto Restart Deadline for Quality Updates** – Set to **2 days**. If you have a compliance requirement for devices to update within a set number of days, then balance deferrals with the configured install and automatic restart deadline. This setting forces the device to restart after the configured deadline interval is reached, which starts after the update is installed. The restart can occur during active hours and does not allow the user to reschedule.

- **Configure Install Deadline for Quality Updates** – set to **0 days**. If you have a compliance requirement for devices to update within a set number of days, then balance deferrals with the configured install and automatic restart deadline. This setting forces the device to install available updates at the configured interval, not waiting for the "Automatic Maintenance" window.

- **Auto-Restart Notification** – Set to **User Dismissal (2).** Windows Update toast notification will not disappear, which prevents issues such as unexpected restarts. However, during the pilot, it is important to test the user experience and durations of notifications combined with Grace Period, Engaged Restart Deadline and Engaged Restart Snooze. For a restart notification flow, see the Windows Update Operational Tutorial on Techzone.

- **Quality Update Deferral** – **Align with deployment rings (number of days)**. A Deferral policy is simply a time period in days to wait before an update is approved for download. The update will install in the next daily maintenance schedule which is out of Active Hours.

- **Allow Microsoft Updates** – Set to **Disabled (0).** This policy scans for and installs Microsoft Application updates. Enable this if not managing application lifecycles of apps such as Microsoft Office or Windows Store apps.

- **Exclude Windows Update Drivers from Quality Updates** – Set to **Disabled (0)** if you want driver updates included in the Windows Update catalog.

- **Install Signed Updates from 3rd Parties** – Set to **Disabled (0)** if not allowing 3rd Party Signed Updates from the Windows Update Catalog.

Note: Microsoft maintains the ability to deploy emergency patches bypassing the approval flow allowing some Updates to install irrespective of approval.

Configure a feature update profile

To set up the ideal configuration, consider the following policies:

- **Target Release Version** – **Set to Feature Update version** such as "21H2". Use if business applications require testing, certification by the vendor, or the OS Update/Lifecycle policy states N-1. Specifies which minor Windows Desktop version to move the device to or stay on until that minor version reaches end of service.

- **Product Version** – Set to “Windows 10” or “Windows 11”. Use if business applications require testing, certification by the vendor, or the OS Update/Lifecycle policy states N-1. Specifies which major Windows Desktop version to move the device to or stay on until that major version reaches end of service.
- **Feature Update Deferral – Set to 0 days if using Target Release Version.** If the testing cycle of feature updates lags behind feature update releases by a number of days, weeks, or months, AND the same deployment rings are used as quality updates, then change this setting to match the delay for that deployment ring.
- **Auto-Restart Deadline For Feature Updates** – Set to **2 days**. This setting forces the device to restart at the configured deadline interval after the feature update is installed. The restart can occur during active hours and does not allow the user to reschedule.
- **Configure Install Deadline for Feature Updates** – set to **0 days**. If you have a compliance requirement for devices to update within a set number of days, then balance deferrals with the configured install and automatic restart deadline. This setting forces the device to install a feature update when available according to the deferral policy, at the configured interval, not waiting for the “Automatic Maintenance” window.

Configure a delivery optimization profile

The following policy settings are recommended to improve the performance and cache hit rate of delivery optimization. Test these settings during your pilot to determine if they suit your environment.

- **Download Mode** – Set to **Use Peers on Same Local Network (2)**.
- **Delivery optimization GroupId** – Set to a **GUID**, this setting is always required. The delivery optimization GroupId can be any GUID generated in either Powershell or another GUID generator such as VMware Policy Builder. It does not have to be the AzureAD Tenant ID.
- **Max Time in cache in seconds (DeliveryOptimization/DOMaxCacheAge)** – Set to **0 (forever)**. Disk space is controlled by the DOAbsoluteMaxCacheSize setting which defaults to 10GB.
- **Minimum content file size in MB enabled to use Peer Caching (DeliveryOptimization/DOMinFileSizeToCache)** – Set to **1 MB**. The default setting is 100MB. Many updates are smaller than 100MB and therefore are not cached.
- **Delay to check for Peer before using CDN in a background download (DeliveryOptimization/DODelayBackgroundDownloadFromHttp)** – Set to **3600 sec**. Ensuring the device waits for peers to appear on the same local subnet prior to falling back to download from CDN will make delivery optimization more effective.
- **Delay to check for Peer before using CDN in a foreground download (DeliveryOptimization/DODelayForegroundDownloadFromHttp)** – Set to **600 sec**. Once again waiting for peers to appear on the same local subnet prior to falling back to download from CDN. This setting should be set to a shorter time as it is triggered by an active/foreground application such as "Check for updates" in Settings > Update & Security.

- **Min download speed to maintain before using CDN (DeliveryOptimization/DOMinBackgroundQoS)** – Set to **64Kb/s**. The default setting is 500KB/s forcing a device to download from peers and CDN in order to maintain 500KB/s.

For more information, see the Policy CSP-Delivery Optimization page.

Create Pause Profiles

Create a Pause Quality Updates profile and a Pause Feature Updates profile to pause the respective updates to assist with troubleshooting updates. Assign the relevant profile only when required and to the relevant Smart Group of devices. The

- **Pause Quality Updates Start Time (Update/PauseQualityUpdatesStartTime)** - Specifies the date and time when the IT admin wants to start pausing the Quality Updates in **YYYY-MM-DD format**. When this policy is configured, Quality Updates will be paused for 35 days from the specified start date.

- **Pause Feature Updates Start Time (Update/PauseFeatureUpdatesStartTime)** - Specifies the date and time when the IT admin wants to start pausing the Feature Updates in **YYYY-MM-DD format**. When this policy is configured, Feature Updates will be paused 35 days from the specified start date.

More information is provided in the Pause and Rollback section below.

Remove previous Windows Update configurations

Before you begin configuring your deployment rings, you need to remove the previous Windows Update configurations from your devices. Clearing previous configurations goes beyond unassigning group policy objects and involves the following steps:

1. **Uninstall SCCM Client (or existing PCLM tool agent)** – If migrating from a legacy PCLM, you need to remove the previous Windows Update configuration from devices. This includes uninstalling the SCCM client. It's important to uninstall the SCCM client prior to removing the registry keys, since SCCM can recreate the settings.

2. **Unassign Windows Update GPO for AD joined devices** – If the device is Active Directory joined and GPOs configured Windows Update previously, unassign the GPO. Keep in mind, this action does not remove the Windows Update settings stored in the registry.

3. **Remove LGPO applied settings** – If a device has an LGPO based policy applied through a build process or Baselines for example, ensure all Windows Update settings are removed from that policy.

4. **Delete Windows Update registry hive or keys** – Delete the registry settings in the Windows Update hive manually, since these setting do not delete on most occasions.
   These settings override the new CSP based settings Workspace ONE deploys, effecting the resulting configuration and manageability of devices.
   See Deploy GPO Removal Sensor section in Techzone article

Configure deployment rings and assignments

Now that you have configured the appropriate profiles, and removed all the legacy PCLM settings, you can create deployment rings. Deployment rings provide control over when updates are released to logical groups of devices. You want to create deployment rings that match the test, validation and release cycles that meet your compliance requirements.

In addition to creating separate quality and feature update profiles, you also want to create matching deployment rings. As discussed above, quality updates and feature updates usually require different deployment schedules with different logical groups of devices and therefore require different deployment rings.

Static assignment

Certain deployment ring types, such as UAT, VIP, and Pilot, should be assigned to a static set of devices. This action allows you to test and validate quality or feature updates in a controlled manner to a known set of devices. There are a few ways to accomplish this goal:

- Tag devices manually with the Tag that is the membership filter in the Smart Group
- Add devices to a Device based Smart Group
- Add the enrolled user into an AD Group with the Active Directory group as the membership filter in the Smart Group.

Dynamic assignment

For the broader deployment rings, you can reduce management overhead with dynamic assignment to logical groups of devices, using a Workspace ONE Sensor and Intelligence automation. Dynamic assignment automatically adds new devices and balances rings when needed. You can use dynamic assignment for quality update, feature update and delivery optimization Smart Groups.

An example "randomizer" sensor assigns all devices a random number between 1 and 10. This number is written to the registry of the device and reported back to Workspace ONE. A Workspace ONE Intelligence automation can then assign a Workspace ONE Tag to the device with the matching Deployment Ring number, which assigns the device to the appropriate profile through the Smart Group membership. **Note**: If using static and dynamic assignment groups, you must exclude all static assignment groups (UAT / Pilot / VIP) from each dynamic assignment group.

It is also possible to utilise a sensor to determine the physical location of a device. The return value could be used to assign a device to a location specific profile. This is particularly useful for environments with WSUS as the download source, where a device will be directed to the closest WSUS server. Two example sensors getGeoLocation1.ps1 and getGeoLocation2.ps1 utilise different methods to determine the location of a device and return a string value.

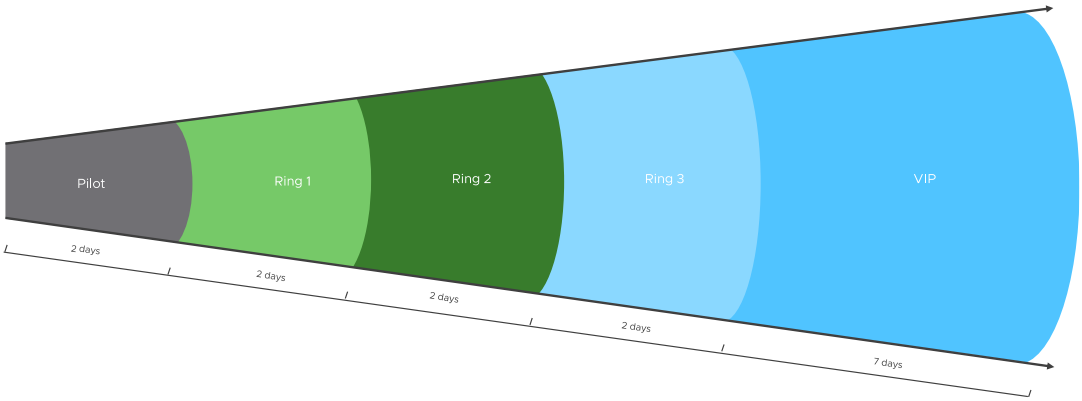
The first sensor uses the NAT'd Internet IP address of the device to determine the approximate location using the IPINFO.IO service. The second sensor uses the Windows Device Location service to determine the Latitude and Longitude of the device and then uses the LOCATIONIQ.COM service to reverse geocode the Latitude and Longitude to provide the street address of the device. Return values include the Country, State,

County, City or Road the device is located.

Quality Updates Deployment Rings

The following example demonstrates the ideal approach for quality update deployment rings. However, Larger customers may require additional deployment rings to stagger updates further or segregate updates to different Lines of Business (LOBs).

The example depicted below staggers each deployment ring by 2 days using deferral. The number of days to stagger or defer, should be based upon the test and validation cycle for quality updates, as well as the time taken for updates to deploy to the majority of devices. A longer deferral equates to a longer 'time to compliance'.



- Pilot Ring – Deploys updates to a static group of users who are members of an Active Directory group or you can manually tag devices.
- Ring 1 to 3 – Assigned dynamically with devices spread evenly across each ring.
    - Ring 1 devices are those that report 1, 2 or 3 from the randomizer sensor. Ring 2 devices are those that report 4, 5, or 6 from the randomizer sensor and Ring 3 devices are those that report 7, 8, 9 and 10. This is just an example of what is possible with WS1 Sensors and WS1 Intelligence Automation. If you cannot use Workspace ONE Intelligence automations, you can manually tag devices or use AD User Groups instead.
    - Pilot and VIP devices are excluded from the assignment
    - Deferral policy used to delay updates - providing time to identify and resolve issues.
    - For additional time, a pause policy can extend the deferral time for subsequent rings.

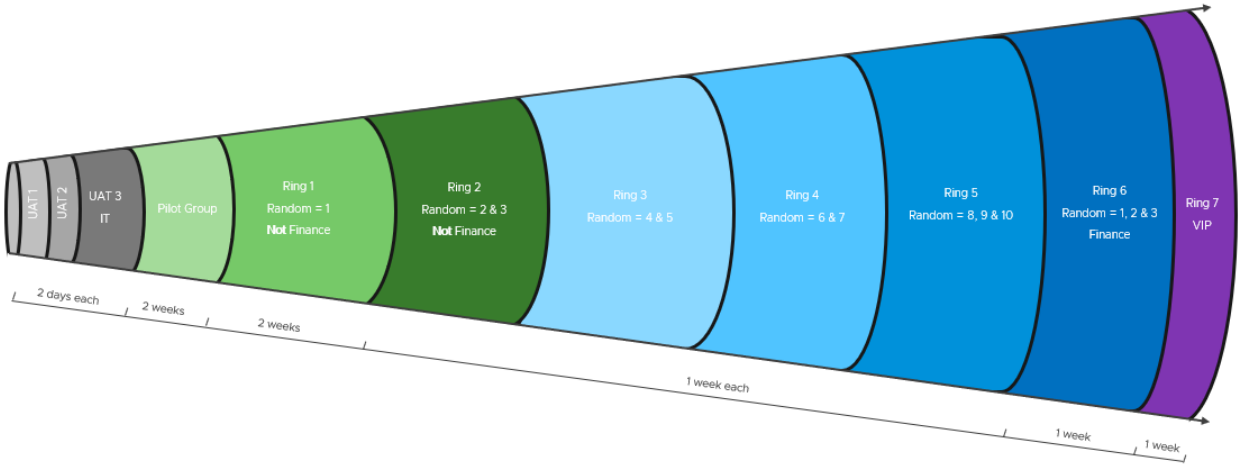
      Note: Urgent and critical updates such as Day-Zero updates usually deploy without delay and therefore without waiting the deferral time.
- VIP – Deploys updates to a static group of users who are members of an Active Directory group. VIP users receive updates last to ensure that all potential issues are resolved.

Feature Updates Deployment Rings

Most feature update deployments are conservative - spreading devices across a greater number of smaller groups within each deployment ring. However, you should base your deployment ring design on balancing your compliance requirements versus your risk profile. These factors dictate the number, size, and membership of your deployment rings.

Additionally, if you have large, multi-site network or if you use WSUS as the download source, then you may require a dedicated deployment ring to seed the cache. This ring would have a small number of dedicated devices in each site. Typically, the pilot deployment ring provides this function.

The following example contains 10 rings that utilize both static and dynamic assignment.

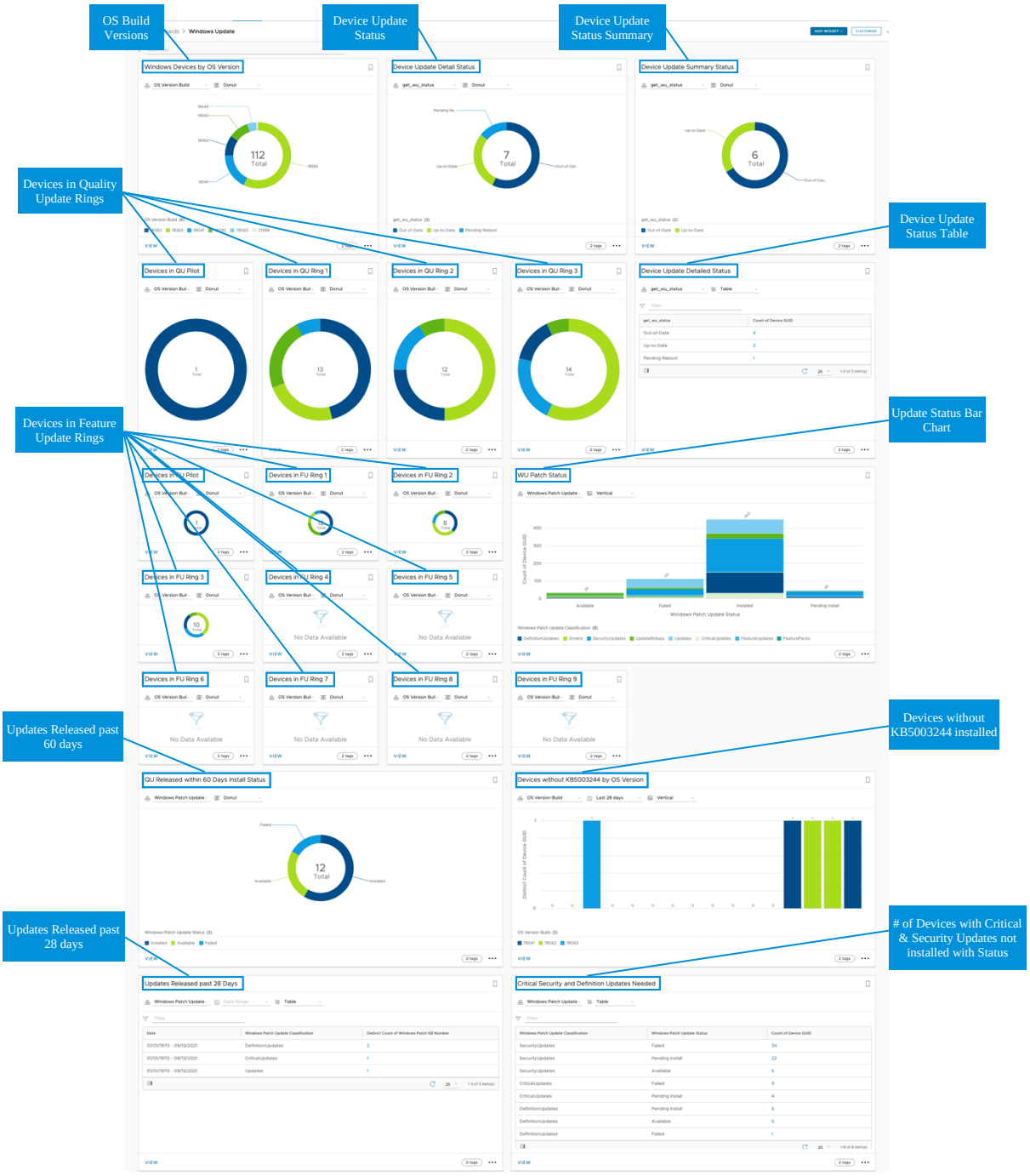


- UAT 1-3 – Deploys updates to a static group of users who are members of an Active Directory group, or you can manually tag devices.
- Pilot Group – Deploys updates to a static group of users who are members of an Active Directory group, or you can manually tag devices.
- Ring 1-6 – Assigned dynamically with each ring equating to between 10% and 30% of the fleet.
    - Ring 1 devices are those that report 1 from the randomizer sensor AND are not in the Finance AD Group. Ring 2 devices are those that report 2 OR 3 from the randomizer sensor AND are not in the Finance AD Group. And so on.
      This is just an example of what is possible with WS1 Sensors and WS1 Intelligence Automation. If you cannot use Workspace ONE Intelligence automations, you can manually tag devices or use AD User Groups instead.
- VIP – Deploys updates to a static group of users who are members of an Active Directory group. VIP users receive updates last to ensure that all potential issues are resolved.

Monitor Compliance with Workspace ONE Intelligence

Lastly, we need to monitor and report on the status of devices across the fleet as well as within each Deployment Ring. We need to see where update installs and restarts are pending or failed, especially in the early deployment rings to resolve issues before broadening deployments further.

You can use Custom Dashboards to focus on Device Update Status as shown in this example:



An example dashboard is available for download here that can be imported into Workspace ONE Intelligence. The example dashboard uses the wu_status sensor for a number of the widgets. Be sure to deploy the sensor 24 hours prior to importing the dashboard to prevent import validation errors.

Device Update Status is the overall Windows Update status of the device rather than the status of an individual update (KB). This widget uses the wu_status sensor to show the following statuses:

- Up-to-Date
- Out-of-Date
- Update-Failed
- Pending Reboot
- No Status

You can also use the dashboards for quick insights into:

- Updates Released in past 28 Days
- Critical Security and Definition Updates Needed
- Devices in Quality Update deployment rings including OS build versions in each deployment ring
- Update Status per Quality Update deployment ring
- Devices in Feature Update deployment rings including OS build versions in each deployment ring
- Quality Updates Install Status Released within past 60 Days
- Devices by OS build versions
- Device Models with OS build versions
- Timeline of OS build versions
- Deployment status of individual patches, e.g. Day Zero Patches

You can also leverage the Vulnerability Dashboards provided within Workspace ONE Intelligence and focus on managing against the greatest vulnerabilities. This is described in https://techzone.vmware.com/meeting-security-slas-through-intelligent-patch-automation-vmware-workspace-one-operational-tutorial.

Operational Considerations

After you create your quality and feature update profiles, configure delivery optimization profiles, and form your deployment rings, the next step is to review the administrative functions for lifecycle management, and other Day 2 operations.

Zero-Day patching

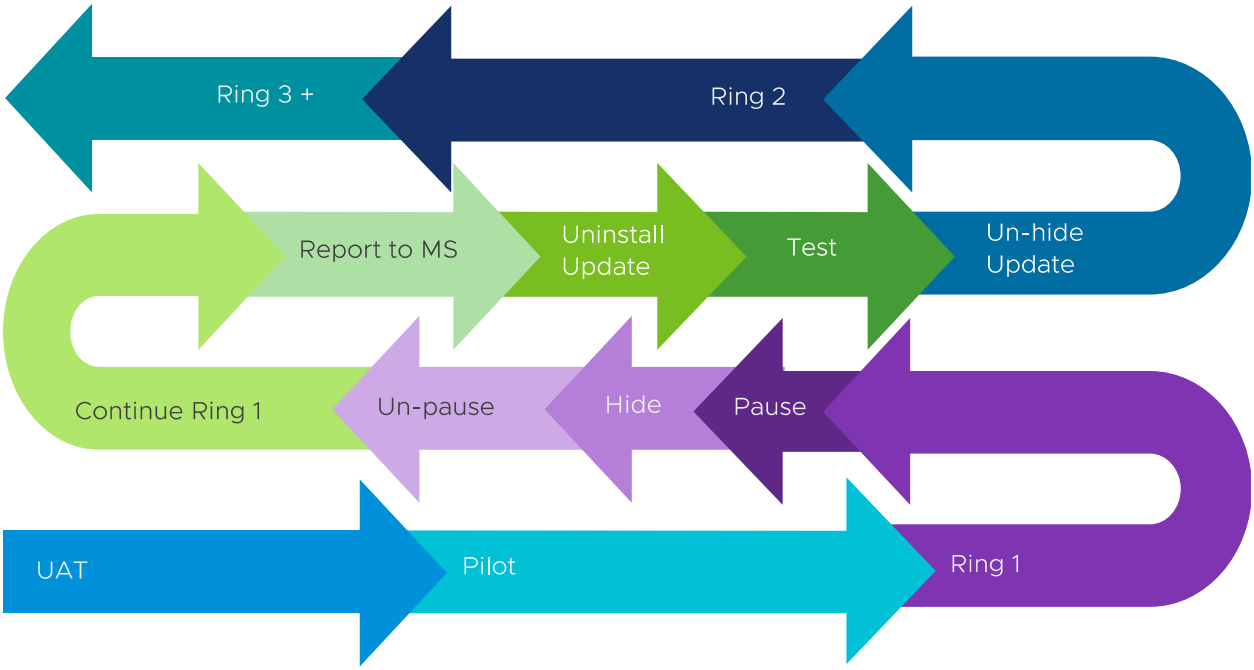
You can deploy an urgent updates to devices is as an application or as a Windows Update. Microsoft often forces deployment of zero day updates bypassing the deferral policy. As such, in most scenarios this step is not required, except to force a device to check for updates and force install the update without waiting for the maintenance window.

Deploying as an application involves packaging and deploying the patch as an application in the Workspace ONE console. It is a little time-consuming to do this, and whilst not supported by VMware, have a look at Brooks Peppin's Windows Updater Tool for Workspace ONE as he has done the hard work.

The other option is to use the PSWindowsUpdate Powershell Module to force install an update. The InstallUninstallWU.ps1 script leverages the PSWindowsUpdate Powershell Module, accepting the update KB number to Install as a parameter, making it easy to deploy as a script or application within Workspace ONE UEM.

Pause and Rollback

The pause and rollback process is similar to managing an update issue with a legacy PCLM, or when manually approving updates.



Here's how it works – an update deploys to the first group (UAT). Then, if no issues are reported, the updates deploy to the next group (Pilot), and so on. If a group reports an issue with an application (Ring 1), or if an install failure appears in the dashboards and reports, then the Administrator pauses the updates for that group and subsequent groups. This prevents the problem update from going out to the next group and allows the Administrator time to troubleshoot and resolve the issue before continuing through the remaining groups of devices.

When an issue is identified, an Administrator can pause all updates until the issue is resolved and a new revision of that update is available, or just hide the bad update. Similar to the InstallUninstallWU.ps1 script, the HideUnhideWU.ps1 script leverages the PSWindowsUpdate Powershell Module and accepts the update KB number to Hide as a parameter, making it easy to deploy as a script or application within Workspace ONE.

On update issues, Microsoft's stance is, it is not about IF you deploy an update, but WHEN you will deploy an update. If there is a problem with an update, Microsoft will rectify it and release a new version of that same update.

For more information, see the Windows Patch Rollback section in TechZone.

Note:

- The Update CSP Rollback node only works for the latest quality and feature updates installed, however there are some significant requirements.
- The Update CSP Rollback node does not uninstall cumulative updates
- There is no Rollback or Uninstall capability in the newer Policy\Update CSP
- The InstallUninstallWU.ps1 script can be used to uninstall individual updates from a device

For 3rd party patches, consider using the uninstall command line parameters from the vendor through Workspace ONE Scripts or as a Workspace ONE application deployment.

Feature updates

Feature updates should be deployed through Windows Update using the Feature Update policy described in the sections above.

Only deploy Feature Updates as an application if it is absolutely necessary. In those cases, keep the following in mind:

- ZIP the full Windows Feature Edition from ISO
- Use the install command: setup.exe /auto upgrade
- Consider using existing deployment rings (Smart Groups) to target devices
- Leverage Peer Distribution (BranchCache) to reduce downloads

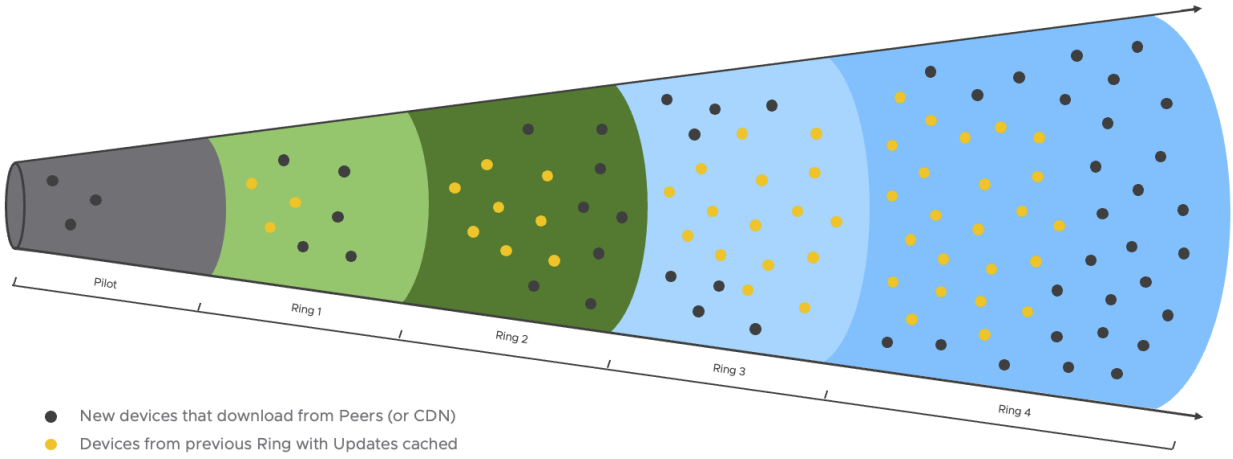
Network

Network monitoring is key to ensuring IT operations do not impact business operations. Part of that is to ensure Windows Update downloads do not overwhelm the network and impact business applications. Delivery optimization is designed to detect congestion through latency, throttle downloads, and restart downloads from the previous chunk so as not to flood the network and download the same data if there are network reliability issues.

In order to prevent untrusted devices peering, use Delivery Optimization GroupID. DOGroupId prevents devices peering with devices that have a different DOGroupId. This allows the use of Download Mode Use Peers on the Same Subnet securely.

Coupled with deployment rings, Delivery Optimization provides a very effective method to reduce update downloads across an internet gateway by first trying to download from Peers. As devices within each Deployment Ring have Updates "approved", the devices will try to connect to a Delivery Optimization peer and download the relevant updates in chunks, quite often downloading chunks from multiple peers simultaneously.

As depicted in the diagram below, these devices then become a Delivery Optimization cache for other devices in the same Deployment Ring or subsequent Deployment rings.



Recommended Settings Summary Table

Additional information on each setting can be found at the Policy CSP-Update and Policy CSP-DeliveryOptimization pages on Microsoft.com.

Setting	CSP Path	Recommended Setting
Quality Update Settings		
Service Channel	Update/BranchReadinessLevel	16
Install Updates Automatically but Let User Schedule Restart	Update/AllowAutoUpdate	1
Active Hours Maximum (Hours)	Update/ActiveHoursMaxRange	Set to total hours
Active Hours (Start)	Update/ActiveHoursStart	Set to start time of business day
Active Hours (End)	Update/ActiveHoursEnd	Set to end time of business day
Wakeup OS for Daily Scheduled Maintenance	Update/AutomaticMaintenanceWakeUp	1
Require Update Approval	Update/RequireUpdateApproval	0
Auto-Restart Notification	Update/AutoRestartRequiredNotificationDismissal	2
Configure Install Deadline for Quality Updates	Update/ConfigureDeadlineForQualityUpdates	0
Auto Restart Deadline for Quality Updates	Update/AutoRestartDeadlinePeriodInDays	2
Auto Restart Deadline Grace Period	Update/ConfigureDeadlineGracePeriod	2
Quality Update Deferral	Update/DeferQualityUpdatesPeriodInDays	number of days aligned to Deployment Ring
Allow Microsoft Updates	Update/AllowMUUpdateService	0
Exclude Windows Update Drivers from Quality Updates	Update/ExcludeWUDriversInQualityUpdate	0
Install Signed Updates from 3rd Parties	Update/AllowNonMicrosoftSignedUpdate	0
Feature Update Settings		
Target Release Version	Update/TargetReleaseVersion	set to Feature Update version
Product Version	Update/ProductVersion	set to OS version
Feature Update Deferral	Update/DeferFeatureUpdatesPeriodInDays	0
Configure Install Deadline for Feature Updates	Update/ConfigureDeadlineForFeatureUpdates	0
Auto Restart Deadline For Feature Updates	Update/AutoRestartDeadlinePeriodInDaysForFeatureUpdates	2
Delivery Optimization Settings		
Download Mode	DeliveryOptimization/DownloadMode	2
Delivery optimization GroupId	DeliveryOptimization/DOGroupId	GUID
Max Time in cache in seconds	DeliveryOptimization/DOMaxCacheAge	0
Minimum content file size in MB enabled to use Peer Caching	DeliveryOptimization/DOMinFileSizeToCache	1

Delay to check for Peer before using CDN in a background download	DeliveryOptimization/DODelayBackgroundDownloadFromHttp	3600
Delay to check for Peer before using CDN in a foreground download	DeliveryOptimization/DODelayForegroundDownloadFromHttp	600
Min download speed to maintain before using CDN	DeliveryOptimization/DOMinBackgroundQoS	64
Pause Profile Settings		
Pause Quality Updates Start Time	Update/PauseQualityUpdatesStartTime	yyyy-mm-dd
Pause Feature Updates Start Time	Update/PauseFeatureUpdatesStartTime	yyyy-mm-dd

Summary and Additional Resources

This document provided guidance and best practices for Windows OS lifecycle management. Procedures included:

- Guidance and planning OS lifecycle management for Windows 10 onwards
- Configuring OS lifecycle management policies for Windows 10 onwards
- Recommended policy settings

For more content on modern management, see the Modernize Desktop Management page on VMware TechZone.

Additional Resources

For more information about Modern Management of Windows with Workspace ONE, you can explore the following resources:

- VMware Professional Services
- VMware Workspace ONE and VMware Horizon Packaging and Licensing guide
- VMware Knowledge Base
- VMware Product Interoperability Matrices

Changelog

The following updates were made to this guide:

Date	Description of Changes
8/20/21	Date created
8/16/22	Updated to include Windows 11, ProductVersion and additional Pause and Rollback considerations
/5/31/23	Update links

About the Author and Contributors

Phil Helmling has been working in IT for over 30 years, with nearly two decades' hands-on experience with VMware technologies, and nearly three decades with Windows technologies. Specializing in IT architecture in the End User Computing space, Phil has designed and deployed hundreds of successful solutions for organizations of all sizes and verticals.

- Phil Helmling, EUC Staff Architect, EUC Engineering, VMware

Feedback

Your feedback is valuable.

To comment on this paper, contact VMware End-User-Computing Technical Marketing at euc_tech_content_feedback@vmware.com.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com.
Copyright ©2022 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents. VMware is a registered trademark or trademark of VMware, Inc. and its subsidiaries in the United States and other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. Item No: vmw-wp-temp-word 2/19