WHITE PAPER
DECEMBER 2017

IT@INTEL

Developing a Gold Standard for Driver and Firmware Maintenance

Our Gold Standard configuration and process for drivers and firmware maintenance allow us to improve the health of our PCs, as well as the user experience and productivity.

Executive Overview

Advances in modern client computing have brought broader features and capabilities to enterprises, helping them develop new markets, innovate new products, and improve communication. But these advancements have also brought new challenges. Solution providers are releasing updates faster and more frequently, and not every update is required for every environment. Microsoft Windows* 10, as well as systems built on Intel® architecture, may require upgrades to drivers and firmware to provide all the capabilities and functionality users expect.

At Intel IT, we developed a Gold Standard configuration for our environment based on our experience and our lessons learned. We determine which drivers and firmware to upgrade based on specific criteria that balance the need to upgrade a platform against the disruption it might cause. We consider the security risk, whether the upgrade fixes known bugs, if the upgrade is required for new features or OS upgrades, and if the upgrade should be included in our standard build. Our process includes:

- Identifying prerequisites
- Testing and deployment
- Monitoring and communication
- Contingency planning

Our Gold Standard configuration and the process for driver and firmware maintenance has allowed us to mitigate many of the problems and vulnerabilities that older, non-optimized drivers and firmware can introduce into the environment, and has also helped Intel to remain focused on security. Overall, we have increased our success rate to more than 90 percent and improved the user experience.

Rocky Tejinder Singh
Client Platform Engineering Manager, Intel IT

Laura Warner
Configuration Manager, Intel IT

Dan Codorean
WLAN Product Engineer, Intel IT

Contents

Contributor

Lelia Barlow
Security Engineer
Intel Software and Services Group

Acronyms

TPM	Trusted Platform Module
UEFI	Unified Extensible Firmware Interface

Unified Extensible Firmware Interface (UEFI)

The Unified Extensible Firmware Interface (UEFI) Specification defines the interface between the OS and platform firmware. UEFI provides a standard environment for booting an OS and running pre-boot applications. The term BIOS is typically used to refer to a specific Intel® architecture firmware implementation, based on older standards and methods. Millions of UEFI-capable systems are in use today, and the vast majority of new PCs are UEFI enabled. The UEFI Forum is a nonprofit industry-standards body where high-tech companies and open source entities work together to advance innovation in firmware technology standards.

Background

The latest Intel® technologies, as well as Microsoft Windows* 10, bring an expanded set of features and capabilities to enterprise computing (Figure 1). But these same advancements in modern computing pose the following challenges to Intel's IT department:

- **Security vulnerabilities.** Increasingly, critical vulnerabilities are being found within lower-level drivers and firmware, requiring urgent remediation. While some of these vulnerabilities can be temporarily managed by disabling OS services, many require driver and firmware updates to fully mitigate the risk.
- **Client health.** We have found it increasingly necessary to assess each update and determine if it is critical and how it will impact device performance. For example, some older devices do not take advantage of newer OS capabilities, and initial releases are not always fully optimized for performance on the latest Intel® architecture.
- **OS requirements.** The OS-as-a-service delivery model enables solution providers to release updates more frequently, but not every update is appropriate for every computing environment. Driver and firmware updates may be required when the OS is upgraded or to take advantage of new Intel architecture capabilities. For example, the camera function may not work as expected in Microsoft Windows 10 unless specific drivers are also upgraded.

To provide Intel employees with PC clients that perform as expected, we continually seek cost-effective ways to manage and maintain the devices. To minimize the performance degradation as the device ages, mitigate security risks, and take advantage of the latest hardware and software features and capabilities, we developed a systematic approach to upgrading the OS, drivers, and firmware across our entire fleet.

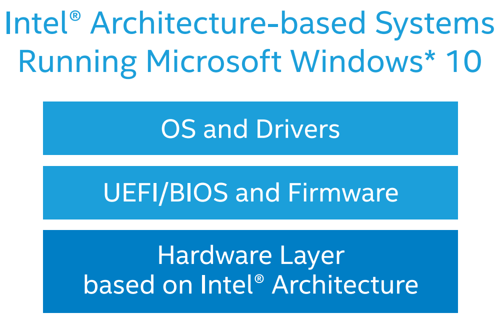

Intel® Architecture-based Systems Running Microsoft Windows* 10
OS and Drivers
UEFI/BIOS and Firmware
Hardware Layer based on Intel® Architecture

Figure 1. When deploying systems, including those based on Microsoft Windows* 10 and Intel® architecture, up-to-date compatible drivers and firmware maximize system stability, security, and performance.

Solution

Though Intel IT routinely updates the drivers and firmware on all supported platforms in our computing environment, we do not use every update provided by the manufacturers and solution providers for every platform. We decide whether to update a platform by balancing the critical nature of the upgrade against the disruption it might cause to the user. We developed a Gold Standard configuration for our environment to help determine which drivers and firmware to upgrade using the following criteria:

- **Critical security risks.** Vulnerabilities identified in low-level drivers and firmware often require urgent remediation. In the past year, we have assessed security vulnerabilities in touchpads, audio drivers, Unified Extensible Firmware Interface (UEFI)/BIOS firmware, Trusted Platform Module (TPM) firmware, and several other firmware solutions.
- **Known bug fixes.** Fixes for problems we experience in Intel's environment, such as those affecting the user experience or build issues, are prioritized above others.
- **New features and capabilities.** We determine whether a driver upgrade is necessary to the function of new software or OS features. For example, deploying Intel® Active Management Technology (Intel® AMT) requires upgrading the Intel® Management Engine (Intel® ME) driver first.
- **OS requirements.** Windows 10 requires driver and firmware upgrades before the OS is upgraded. We primarily use internal certification labs to identify and document incompatibilities. We are also taking advantage of the recently announced Windows upgrade analytics capability to further accelerate our process.
- **Standardization.** As part of our Gold Standard configuration, we establish standards for required drivers and firmware, and transition client models to a common, tested, healthy version to reduce the complexity of the environment.

Critical Response in the Modern Environment

Today's enterprise computing environments require more frequent updates with more varied dependencies. Sometimes, unplanned, critical updates must be deployed in days, rather than weeks. Intel IT balances risks with velocity by considering the following:

- **Risk analysis.** Known vulnerabilities are analyzed and evaluated to determine the risks to Intel and how to mitigate them. In many cases, patching is the only mitigation, and at that time we examine the impact of the patch on users. For example, we recently experienced a Trusted Platform Module (TPM) vulnerability in our environment affecting up to 55,000 devices, but the patch involved two or three reboots during the update process, which led to user frustration. By working with our OEMs and security teams, we have been able to move to a single reboot solution for the majority of the clients affected by the TPM issue.
- **Velocity.** Responding to critical updates requires an agile release process to quickly evaluate the health of a package before releasing it. We define levels of velocity to manage critical updates using the same process we use for all updates, but in an accelerated manner. This sometimes requires making an educated guess about the risks of rapid deployment. For non-critical patches, Intel IT uses a long release cycle of 11 weeks to ensure the highest performance and lowest risk for our users. For more urgent levels of vulnerability and risk, we have defined different cycle times.
- **Dependency mapping.** We map dependent drivers and firmware, and then sequence those updates before package deployment. Planned upgrades take into account the patching sequence, bundling, and relevancy of the patches and their dependencies. For example, a graphics patch may depend on dock, Unified Extensible Firmware Interface (UEFI), or other drivers in the environment. We occasionally create complex relevancy maps to verify that all driver and firmware dependencies are met prior to deployment.

We currently manage over 300 individual drivers across more than 30 platforms. Driver and firmware patches remediate 2,000 to 100,000 machines, depending on their use. With a robust patching and self-healing methodology that balances the health of the platform with the risks associated with the change, we have experienced a 90 percent or higher success rate.[1] With Microsoft Windows Analytics* and internal testing processes, we can identify incompatibilities and release patches prior to Windows 10 OS upgrades to ensure minimal disruption to user productivity.

>90% SUCCESS RATE

with our robust patching and self-healing methodology.[1]

To meet and maintain our Gold Standard configuration, we evaluate the update options, pre-requisites, the technical approach, the business processes (which includes change management), and contingency planning. We complete all upgrades by identifying lessons learned, which are incorporated into the process for future upgrades.

Prerequisites

It is important to determine whether the upgrade will work for the targeted systems and make necessary adjustments to the deployment package. Occasionally, a patch requires bundled components, such as:

- **Encryption suspension.** UEFI and other firmware updates may cause unexpected interactions with full disk encryption solutions. We use a variety of encryption solutions, each of which requires unique steps to regulate interactions and provide a good user experience. We work with solution providers and security teams to address sensitivities in firmware releases and deploy specific firmware update tools and packages for each system. Modern computing platforms often have three or more firmware components that must be maintained, and some system-specific update tools run independently. Data may be vulnerable during this phase, and we take steps to minimize the risk by limiting how long encryption is suspended.
- **Bundled patches.** Occasionally, we need to patch multiple components at the same time and in sequence to provide proper installation and to minimize disruption to the user. For example, bundling TPM firmware and UEFI is required for TPM firmware updates.
- **Preservation of non-standard features.** While deploying wireless updates, for example, we consider various technologies that may be impacted, such as wireless displays and location awareness. We preserve the existing configuration when updating drivers.
- **Post-deployment optimization.** We generally prefer to deploy drivers with their default configurations. However, in some cases, a post-install configuration is required to address specific needs. This includes optimizing wireless drivers used in real-time applications, such as web conferencing.

[1] Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

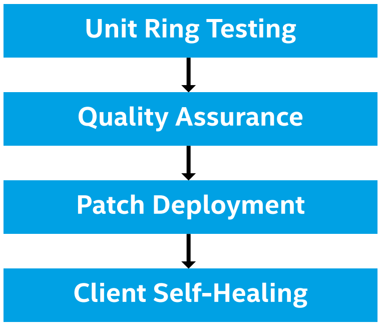
With commercially available configuration management tools, understanding the tool's capabilities and how to use them is essential. It is also important to consider how users will react to the upgrade and whether we can measure success rates. Prior to testing and deployment, we identify the number of clients impacted and the experience they might encounter, which is necessary to make decisions about the type of solution we offer. Third-party patching tools often detect about 90 percent of affected clients, and a 5-15 percent gap can lead to extra time optimizing drivers during post-deployment that we must include in planning.

Technical Approach

Driver and firmware packages undergo basic validation testing for functionality in our engineering labs. After testing, we use a technical pipeline that includes the following steps:

1. **Unit ring testing.** Unit ring testing encompasses small groups of users who participate in pilot deployments. We monitor events in the background and collect user feedback. With this unit ring data, we can evaluate the impact of the change in the environment, such as unexpected shutdowns or fatal errors. This also allows us to target known problems and determine whether an upgrade corrects the problem or improves performance.
2. **Quality assurance.** We formally validate the package as part of the Information Technology Infrastructure Library (ITIL) process by testing the latest OS release on new devices to verify functionality of the driver or firmware and the applications dependent on it.
3. **Patch deployment.** We phase patch deployments over several weeks. During week one, we deploy patches to approximately 250 users in each geographic region. Product engineers and product owners monitor patch deployment and incident rates to confirm success. During week two, the patch is deployed to approximately 5,000 clients in each region. In week three, we deploy the patch to all relevant users. Weekly monitoring continues to measure success rates until the patch has been successfully deployed to 90 percent of the total client population, after which the patch continues to circulate for a few months to catch any machines that were offline.
4. **Client self-healing.** When we cannot update by patching or the device fails, users can evaluate the device on their own and fix the issue through our self-healing service. We send push notifications to inform users of the required action and provide instructions on how to take that action. In addition, clients include an application to self-evaluate health and take remediation steps.

Technical Approach for Upgrading Drivers and Firmware



Business Process

The business process and change management phases run in concert with the technical approach and are an important part of determining the fitness of the patch (see Figure 2). We use a standard process for all upgrades, regardless of whether it is a driver, firmware, or other system update.

Success Tracking and Change Management

Tracking success is more than testing; it includes careful evaluation of the test results before deployment, as well as the overall success of the deployment. Our deployment team includes a dedicated change manager who reviews the unit ring testing results from Step 1, the quality assurance results from Step 2 and the product data, then assigns an overall risk rating to the upgrade.

The team also includes event managers, configuration managers, and problem managers who use the results of the data to make recommendations about whether the deployment is a "Go" or "No Go." In one No Go scenario, for example, we were planning a firmware upgrade with several critical fixes that we believed would lower the number of unexpected shutdowns on some platforms when waking up from the sleep and hibernate states. Our initial testing passed, but some clients experienced severe audio problems during web conferencing sessions while connected to a specific dock model. Though it impacted a small number of clients, we decided not to deploy the package until the solution provider corrected the issue.

Once we reach Go status, we prepare the release for deployment, include it in the standard OS build, and create self-healing scenarios.

Technical Support and Communication

Once the upgrade is a Go, we involve the support team and user base through the following activities:

- **Notify support.** We alert our support staff about the upgrade and inform them of the possible issues and the subsequent reports they may receive.
- **Update support documents.** We update relevant knowledge articles and troubleshooting guidelines with the latest information.
- **Notify the user base.** We communicate the upcoming change to the user base and who to contact if they experience issues. We also include a splash screen with information about what to expect and any important instructions. For example, some upgrades require that the user remain connected to AC power, and that is communicated on the splash screen. We have seen a significant reduction in failure rates since implementing this process.

Technical Approach and Business Process for Upgrades

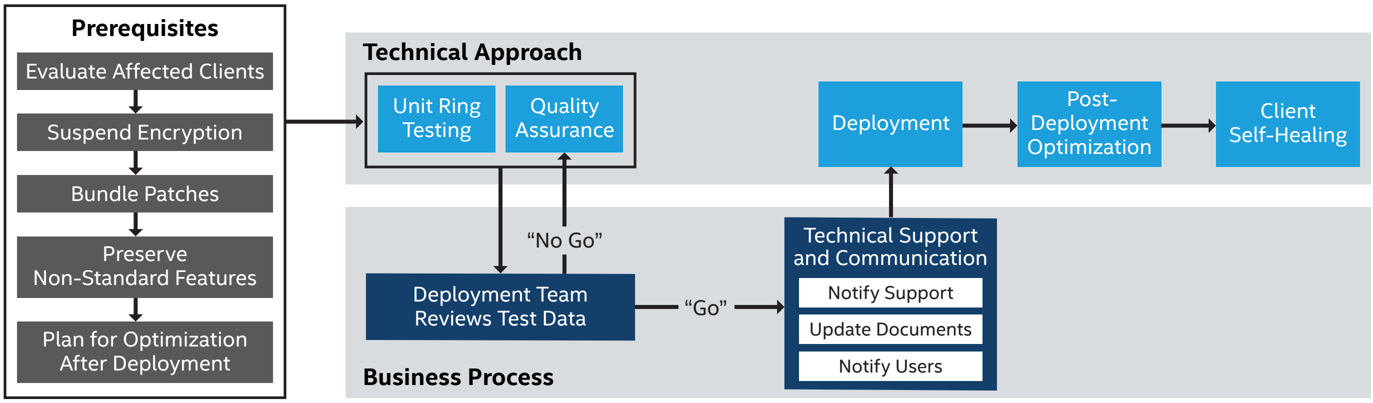


Figure 2. The business process and change management phases run parallel with the technical approach and are important for determining the fitness of the patch.

We carefully monitor the upgrade rollout for failures and determine if adjustments are necessary. We also maintain continuous communication with our technical support staff and the user base about any unforeseen issues that arise.

Contingency Planning

Our driver and firmware maintenance process is built on what we have learned through years of experience. Our Gold Standard configuration is the result of dependency mapping, success tracking, close communication with suppliers, and testing. Occasionally, a patch still fails. When this happens, we strive to understand the success metrics through our unit ring testing; monitor for critical and catastrophic failures during deployment and adjust in real time; communicate the resolution; and use self-healing options where appropriate. It is rare in today's modern computing environment that an update permanently damages a computer, but it is still critically important to have processes in place prior to patching.

At Intel, everyone on the operations team, including IT service center leads, can temporarily stop a patch to limit damage if there is a high number of reported problems. Product owners are immediately notified and begin debugging the issue, and we conduct trend analyses to better understand the impact. If the problem can be solved with a package or patch relevancy adjustment, those changes are made, revalidated, and we begin the deployment cycle again.

As we deploy an update, we monitor the environment using Microsoft Windows Event Forwarding, collecting information on unexpected shutdowns and blue-screen errors as well as incident generation. The information collected is evaluated throughout all deployment phases. During unit ring testing, for example, if a small number of clients is affected by an issue, we often use Windows Management Instrumentation (WMI) to obtain remote system information. If necessary, we also work with our solution providers to understand the issues they have encountered in other environments.

For critical patches, we use an accelerated cycle time that adheres to the process, but allows for rapid deployment (see Figure 3).

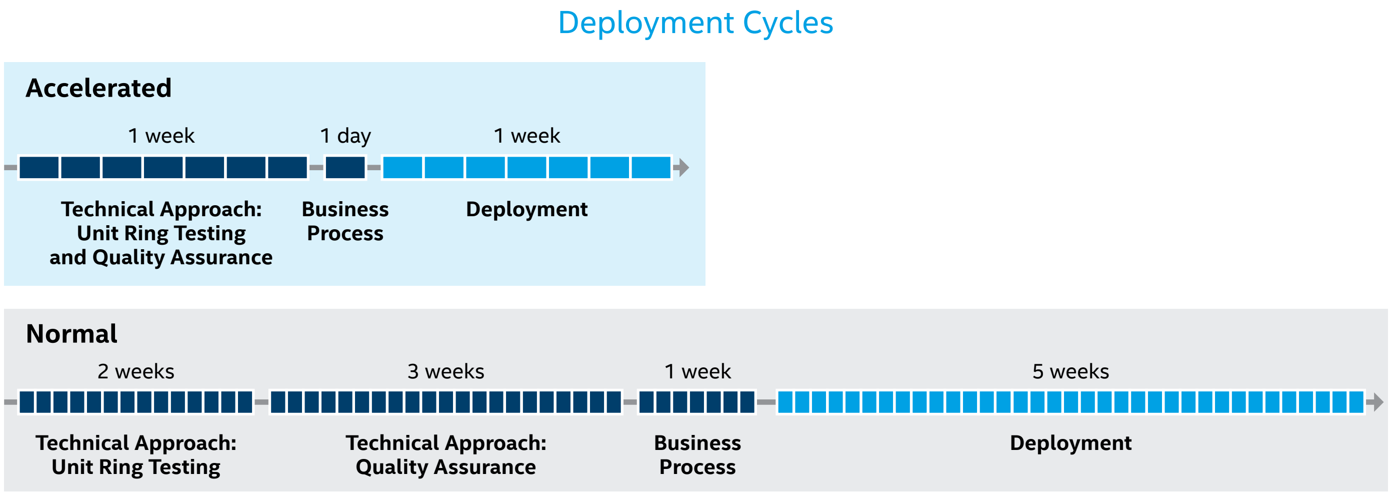

Deployment Cycles
Accelerated
1 week
1 day
1 week
Technical Approach: Unit Ring Testing and Quality Assurance
Business Process
Deployment
Normal
2 weeks
3 weeks
1 week
5 weeks
Technical Approach: Unit Ring Testing
Technical Approach: Quality Assurance
Business Process
Deployment

Figure 3. The deployment cycle time is accelerated when the patch is critical for vulnerabilities in the environment.

Conclusion

Modern computing brings greater capabilities to the enterprise, but it has also introduced some challenges in ongoing management and security. Low-level drivers and firmware must be updated to mitigate security vulnerabilities. The velocity and cadence of Windows 10 updates, in addition to the new generation of firmware and drivers—including those that take advantage of new Intel technologies—make it necessary to update and test low-level drivers to verify compatibility and security, as well as maintain a stable environment that uses these new capabilities.

Based on our experience and lessons learned, Intel IT has developed a Gold Standard configuration for maintaining drivers and firmware, using a clearly defined process. By identifying the prerequisites; thoroughly testing patches; reviewing the results data; bundling and sequencing driver and firmware upgrades; and managing the risks, as well as the deployment, we have increased our success rate to more than 90 percent and improved the user experience.

For more information on Intel IT best practices, visit **intel.com/IT**.

IT@Intel

We connect IT professionals with their IT peers inside Intel. Our IT department solves some of today's most demanding and complex technology issues, and we want to share these lessons directly with our fellow IT professionals in an open peer-to-peer forum.

Our goal is simple: improve efficiency throughout the organization and enhance the business value of IT investments.

Follow us and join the conversation:

- Twitter
- #IntelIT
- LinkedIn
- IT Center Community

Visit us today at **intel.com/IT** or contact your local Intel representative if you would like to learn more.

Related Content

If you liked this paper, you may also be interested in these related stories:

- Advancing the User Experience with Intel® Architecture-based Laptops and Microsoft Windows* 10 paper
- Intel® Product Security Center

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer, or learn more at intel.com.

THE INFORMATION PROVIDED IN THIS PAPER IS INTENDED TO BE GENERAL IN NATURE AND IS NOT SPECIFIC GUIDANCE. RECOMMENDATIONS (INCLUDING POTENTIAL COST SAVINGS) ARE BASED UPON INTEL'S EXPERIENCE AND ARE ESTIMATES ONLY. INTEL DOES NOT GUARANTEE OR WARRANT OTHERS WILL OBTAIN SIMILAR RESULTS.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS AND SERVICES. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS AND SERVICES INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

1217/JDEL/KC/PDF