National Security Agency/Central Security Service

Information Assurance Directorate

Deploying Signed BIOSes to Enterprise Client Systems

November 16, 2012
Revision 1

A product of the Network Components and Applications Division

ADF-2012-1215

Contents

List of Tables

Disclaimer

This Guide is provided "as is." Any express or implied warranties, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the United States Government be liable for any direct, indirect, incidental, special, exemplary or consequential damages (including, but not limited to, procurement of substitute goods or services, loss of use, data or profits, or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this Guide, even if advised of the possibility of such damage.

The User of this Guide agrees to hold harmless and indemnify the United States Government, its agents and employees from every claim or liability (whether in tort or in contract), including attorneys' fees, court costs, and expenses, arising in direct consequence of Recipient's use of the item, including, but not limited to, claims or liabilities made for injury to or death of personnel of User or third parties, damage to or destruction of property of User or third parties, and infringement or other violations of intellectual property or technical data rights.

Nothing in this Guide is intended to constitute an endorsement, explicit or implied, by the U.S. Government of any particular manufacturer's product or service.

Trademark Information

This publication has not been authorized, sponsored, or otherwise approved by Dell, Inc. or Hewlett-Packard Company.

OptiPlex®, Latitude®, and Precision® are registered trademarks of Dell, Inc.

EliteBook® and ProBook® are registered trademarks of the Hewlett-Packard Company.

Microsoft®, Windows®, and Active Directory® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

1 Introduction

The BIOS is low level software, built into the computer's hardware, which is the first code run by a computer when it is powered on. The BIOS initializes the system, and its devices, and then hands off execution to the operating system to finish the boot process[1].

Signed BIOSes are BIOSes that contain digital signatures, can be authenticated with a public key, and have built-in write protection that prevents unauthorized BIOS modifications. These BIOSes are designed and implemented to meet the requirements of NIST Special Publication[2] 800-147[3].

Numerous security researchers have presented information and techniques on abusing and attacking BIOSes at multiple security conferences since 2006[4,5,6,7,8,9,10,11]. While not widely known or commonly understood, malicious BIOSes can be especially devastating since they function at a level below the operating system and are not directly detected by antivirus software. A malicious BIOS may also be capable of reinfecting an operating system after it has been reinstalled and it may be very difficult to completely remove the infection.

Recent malware, such as Mebromi[12] and Niwa[13], installed a malicious BIOS as part of its infection. Systems with signed BIOSes would have prevented Mebromi from installing its malicious BIOS since the malicious BIOS was unsigned. By updating systems to use signed BIOSes, users can mitigate the threat posed by this class of malware that can be very difficult to detect and remove.

This guide is meant to assist United States government and Department of Defense Windows system administrators deploy BIOSes to their enterprise client systems that support signed BIOSes and signed BIOS update mechanisms but do not have signed BIOSes installed by default due to these systems predating the NIST SP 800-147 standard. Vendors that implement signed BIOses currently ship systems with a signed BIOS already installed. This guide also provides information on tools for managing BIOSes that are freely available and officially supported by vendors for commercial use. The guide assumes administrators operate in a restrictive network environment where common remote management protocols may be blocked and common automation technologies may be disabled. Very basic techniques and technologies are used in this guide to apply to the widest audience possible and to allow easier integration into restrictive environments.

---

[1] Basic Input Output System. http://en.wikipedia.org/wiki/BIOS
[2] NIST Special Publications (800 Series). http://csrc.nist.gov/publications/PubsSPs.html
[3] NIST SP800-147: Basic Input/Output System (BIOS) Protection Guidelines. http://csrc.nist.gov/publications/nistpubs/800-147/NIST-SP800-147-April2011.pdf
[4] Implementing and Detecting an ACPI BIOS Rootkit. http://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-Heasman.pdf
[5] Firmware Rootkits: The Threat to the Enterprise. http://www.nccgroup.com/secure/PHunkzRGsTI%3d/1099/
[6] Hacking the Extensible Firmware Interface. http://www.nccgroup.com/secure/sROmcvPNQ04%3d/1099/ or https://www.blackhat.com/presentations/bh-usa-07/Heasman/Presentation/bh-usa-07-heasman.pdf
[7] Persistent BIOS Infection. http://www.coresecurity.com/files/attachments/Persistent_BIOS_Infection_CanSecWest09.pdf
[8] Deactivate the Rootkit. http://www.blackhat.com/presentations/bh-usa-09/ORTEGA/BHUSA09-Ortega-DeactivateRootkit-SLIDES.pdf and http://www.blackhat.com/presentations/bh-usa-09/ORTEGA/BHUSA09-Ortega-DeactivateRootkit-PAPER.pdf
[9] Attacking Intel BIOS. http://www.blackhat.com/presentations/bh-usa-09/WOJTCZUK/BHUSA09-Wojtczuk-AtkIntelBios-SLIDES.pdf
[10] A Real SMM Rootkit. http://www.phrack.org/issues.html?issue=66&id=11#article
[11] Hardware backdooring is practical. http://www.toucan-system.com/research/blackhat2012_brossard_hardware_backdooring.pdf
[12] Mebromi: the first BIOS rootkit in the wild. http://blog.webroot.com/2011/09/13/mebromi-the-first-bios-rootkit-in-the-wild/
[13] 'Bioskits' Join Ranks of Stealth Malware. http://blogs.mcafee.com/mcafee-labs/bioskits-join-ranks-of-stealth-malware

2 Implementations

HP and Dell implement signed BIOSes and signed BIOS update mechanisms for specific desktop and laptop models that are commonly used in enterprises. HP and Dell currently ship enterprise client systems with a signed BIOS already installed. Administrators do not need to take any actions on these systems. Contact vendor sales representatives to find out if a specific current generation model ships with a signed BIOS. Since the original publication of this guide in July 2012, other vendors, such as Samsung[14], have also started shipping models that have NIST SP 800-147 compliant BIOSes.

Previous generations of HP and Dell systems, and all other vendor's systems, did not ship with signed BIOSes installed by default since those systems predated NIST SP 800-147. Currently only Dell and HP provide signed BIOSes for previous generations of their systems. Once a signed BIOS is installed and enabled on these systems, an administrator will be unable to revert to previous versions of BIOSes that are unsigned. This also prevents malicious software from installing a malicious BIOS.

The DoD CIO[15] and DHS[16] have issued memoranda requiring new procurements of PC client systems to have NIST SP 800-147 compliant BIOSes to the maximum extent that is practical. The DoD CIO memorandum also mentions an update to DoD Instruction 8500.02[17] which includes requirements for systems to comply with NIST SP 800-147. In addition to purchasing new systems with signed BIOSes, installing and enabling signed BIOSes on older systems is recommended when a vendor provides signed BIOS updates for older systems.

2.1 Dell

Dell added a feature to the BIOS called Signed Firmware Update. This feature is not enabled by default on previous generations of Dell models so an administrator *must* enable this feature first if they want to enforce the signed update mechanism on older systems. The BIOS version that adds the feature varies based on the specific Dell model. For example, on an OptiPlex 990 the A05 BIOS version added the feature to the BIOS. The first actual signed BIOS version was A06. Enforcement of the A06 BIOS version's signed update mechanism is not enforced until the feature is enabled. On an OptiPlex 755 the A20 BIOS version added the feature and was also the first signed version. If the signed update feature is enabled, then an administrator will be unable to revert to version A19. An administrator will only be able to install version A20 again since it is currently the only signed BIOS version for that model.

2.1.1 Manually Detecting and Enabling the Signed Firmware Update Feature

Once the BIOS version that adds the Signed Firmware Update feature has been identified (see Table 1), install it and let the system reboot to allow the BIOS update to complete. During the next reboot enter the BIOS setup dialog by pressing F2 or F12 when the Dell logo displays during boot. To detect the presence of the feature, start by navigating to the **Security** section. Below that section there should be a

[14] Samsung Series 7 Slate with NIST SP 800-147 Compliant BIOS. http://www.samsung.com/us/computer/tablet-pcs/XE700T1A-A09US
[15] DoD CIO Memorandum: Implementation of Basic Input/Output System (BIOS) Protection Guidelines. http://dodcio.defense.gov/Portals/0/Documents/Signed_Memo_NII001001-11[1].pdf
[16] FISM 12-01: Protected BIOS for New Procurements of Desktop and Laptop Computers. http://www.dhs.gov/xlibrary/assets/nppd/fism12-01-signed.pdf
[17] DoD Instruction 8500.02. http://www.dtic.mil/whs/directives/corres/pdf/850002p.pdf

new entry labeled **Signed Firmware Update**. Select that entry and then look in the right-hand pane to find an option labeled **Enable Signed Firmware Update**. The description of the feature is:

> *This feature, when enabled, will enforce verification of digital signatures in the firmware update payload prior to performing an update of the firmware. Once enabled, the system BIOS cannot be updated to any revision that DOES NOT contain a valid digital signature. Note: You will not be able to change the setting once the feature is enabled.*

Select the option labeled **Enable Signed Firmware Update** to enable the feature. Save the changes and exit the BIOS setup dialog. The next time an administrator enters the BIOS setup dialog and views the Security section, the Signed Firmware Update entry is no longer displayed. Since the feature can only be enabled once and can't be disabled, the Signed Firmware Update entry is removed from the Security section of the BIOS menu once the feature is enabled.

Once an administrator enables the feature, saves the changes, and exits the BIOS setup dialog, they can enter the BIOS setup dialog again and check for evidence that the feature is enabled. Find the BIOS section labeled **System Information**, **System Info**, or **System Board**. Select that section and look in the right-hand pane for the text **Signed Firmware Update Enabled**. This text will be displayed below the BIOS version, service tag, express service code, and asset tag information when the feature is enabled.

If the Signed Firmware Update feature is not enabled, then an administrator can still install a signed BIOS version as they would have installed any unsigned BIOS update in the past, but the signed update mechanism will not be enforced. This means an administrator can still roll back to previous unsigned BIOS versions. When the administrator has the feature enabled they will not be able to roll back to any previous unsigned BIOS versions.

As previously mentioned, current Dell enterprise desktops and laptops ship with signed BIOSes installed by default. Administrators only need to enable the signed BIOS feature on previous generations of Dell enterprise clients.

2.1.2 Automatically Detecting and Enabling the Signed Firmware Update Feature

Dell's Client Configuration Toolkit (CCTK)[18,19] can detect and enable the Signed Firmware Update feature. Different methods can be used to detect and enable the feature depending on which version of CCTK is being used.

2.1.2.1 CCTK 2.0.1 and Earlier Versions

Use the following CCTK command to detect if the feature is supported by the BIOS:

> **cctk.exe --istokenactive=0x0325**

[18] Dell Client Configuration Toolkit documentation. http://support.dell.com/support/edocs/SOFTWARE/smcctk/
[19] Dell Client Configuration Toolkit. http://en.community.dell.com/techcenter/systems-management/w/wiki/dell-client-configuration-toolkit.aspx

The command results in 1 of 4 outcomes:

- Case 1: '**The requested CMOS token 0x0325 is active**' when the feature exists *and* is enabled. The CCTK process return code is **0**.
- Case 2: '**The requested CMOS token 0x0325 is not active**' when the feature exists and is *not* enabled. The CCTK process return code is **0**.
- Case 3: '**The state byte is not available on this system**' when the feature does not exist. The CCTK process return code is **81**.
- Case 4: Any non-zero return code other than 81 should be interpreted as a failure to enable the feature. While this case is possible, it has not been observed.

Unfortunately CCTK may not always detect the feature even if it is supported by the BIOS. Case 2 or 3 may occur on some BIOS versions that support enabling the feature but do not support detecting the feature. This issue is model and BIOS version specific. See the Known Issues section for more information. Despite this limitation for detecting the feature correctly, the feature can still be enabled by using CCTK.

Use the following CCTK command to enable the feature:

cctk.exe --token=0x0325

If a BIOS password, also referred to as the setup password in Dell documentation, is set on the system, then CCTK can't enable the feature unless the password is specified with the valsetupwd option. Use the following CCTK command to enable the feature when a BIOS password is set:

cctk.exe --token=0x0325 --valsetuppwd=*password*

Regardless of which command is used, it results in 1 of 3 outcomes:

- Case 1: '**The requested CMOS token 0x0325 is set**' when the feature was successfully enabled. The CCTK process return code is **0**. If the feature is already enabled and the command is run again, then the behavior is the same. This behavior may lead to a false positive where an administrator or script can't tell the difference between enabling the feature for the first time versus running the command again when the feature has already been enabled. That is why the istokenactive command should be used first to determine if an attempt to enable the feature is even needed. Unfortunately CCTK does not always detect the feature correctly so a scripting process may just have to enable the feature without checking if it is supported first.
- Case 2: '**The state byte is not available on this system**' when the feature is not supported. The CCTK process return code is **81**. In a scripting process, this case shouldn't happen often since if the istokenactive command indicates the feature is not supported, then an administrator or script shouldn't attempt to enable the feature. Due to CCTK not always being able to detect the feature correctly, this may be reported on some models' BIOSes that support the feature.
- Case 3: Any non-zero return code should be interpreted as a failure when enabling the feature and the system should be marked as not successfully completed. Administrators should follow

up with a manual investigation of the system. Many password related error codes fit into this case. See the CCTK documentation for error code numbers[20] related to passwords.

Only the command line version of CCTK can enable the feature in versions 2.0.1 and earlier versions. The CCTK Self-Contained Executable (SCE) feature does not support enabling the feature until CCTK version 2.1.

2.1.2.2 CCTK 2.1 and Later Versions

CCTK 2.1 can use the same methodology as discussed above but it also has a command line option added specifically for enabling the Signed Firmware Update feature. Use the following CCTK command to detect the feature:

cctk.exe --sfuenabled

The output of the command will contain **sfuenabled=yes** when the feature is enabled **sfuenabled=no** when the feature is not enabled. Use the following CCTK command to enable the feature:

cctk.exe --sfuenabled=yes

Use the following CCTK command to enable the feature when a BIOS password is set:

cctk.exe --sfuenabled=yes --valsetuppwd=password

CCTK 2.1 and later versions also have the advantage of being able to deploy the setting using the Self Contained Executable (SCE) feature of CCTK while earlier versions of CCTK can't use the SCE feature to deploy the setting.

If the BIOS does not support the Signed Firmware Update feature, or if CCTK can't correctly detect the feature correctly, then CCTK returns the following message:

'This option is not available or cannot be configured through this tool: sfuenabled'

The CCTK process return code is set to 0 which is missing since the command is not valid.

CCTK may not display the sfuenabled option on some systems, such as the D630 and E6420, even when they have a capable BIOS installed because CCTK may not be able to detect the feature correctly. In some cases, a later BIOS version, such as A13 for the E6420, may allow CCTK to detect the feature correctly. Using the istokenactive and token options in an enterprise deployment is recommended to make the deployment as uniform as possible across all different models and to avoid issues where the sfuenabled option may not be available.

2.2 HP

Installing an HP BIOS version that is a signed BIOS version is the only method of enabling signed BIOSes on previous generations of HP systems. There is no specific feature to enable which makes enabling

[20] Dell Client Configuration Toolkit Error Codes. http://en.community.dell.com/techcenter/systems-management/w/wiki/1953.aspx

signed BIOSes on HP enterprise clients the same as deploying any other HP BIOS update. There is also no way of telling if a signed BIOS is installed other than knowing which BIOS version was the first signed version for a particular model.

3 Model Support

Conformance to the April 2011 version of NIST Special Publication 800-147 requires client desktop and laptop systems to support signed BIOSes. Vendors support signed BIOS implementations on modern business class desktops and laptops that have been released within the last 3-5 years. Vendors may also support signed BIOSes on servers once the NIST SP 800-147B standard for servers is released.

3.1 Dell

Dell supports signed BIOSes on recent OptiPlex, Precision, and Latitude models. Current Dell systems ship with signed BIOSes installed and enabled by default. Previous generations of these Dell models predated NIST SP 800-147 and did not ship with signed BIOSes installed and enabled by default. Some of these models support NIST SP 800-147 starting with a specific BIOS version. See Table 1 for specific models and BIOS versions. The First Signed column in the table denotes the first available BIOS version that is signed. The Feature Added column denotes the first BIOS version that adds the ability to enable the Signed Firmware Update feature. Some Dell BIOS updates have a hard dependency on a previous BIOS version that requires the previous BIOS version to be installed before updating to the latest BIOS version. Table 1 denotes this case with an asterisk next to the BIOS version that is the hard dependency.

In some cases the first signed BIOS version and the BIOS version that adds the Signed Firmware Update feature are the same BIOS version. This is especially common in older models. Newer models may have had the feature added in one BIOS version and the first signed BIOS version as a later BIOS version. Rows in italics denote models where the First Signed and Feature Added columns are different BIOS versions.

The Test Result column denotes if CCTK can successfully detect *and* enable the Signed Firmware Update feature for that particular model and BIOS version. The column's value is:

- '**P**' when CCTK can successfully detect *and* enable the feature.
- '**F**' when CCTK could not detect the feature, could not enable the feature, or there was some other error condition encountered. The number after the letter corresponds to a test case in the Testing Methodology section.
- empty when a test has not been performed on that model yet.

See the Testing Methodology section for information about the testing procedures and the Known Issues section for information about problems that were encountered. Despite the number of test failures listed in the Test Result column in the table below, almost all the failures are *not* critical failures that would prevent an organization from deploying Dell signed BIOSes.

Model	First Signed	Release Date	Feature Added	Release Date	Test Result
OptiPlex 360	A06	11/20/2011	A06	11/20/2011	

Model	First Signed	Release Date	Feature Added	Release Date	Test Result
OptiPlex 380	A06	10/27/2011	A06	10/27/2011	
OptiPlex 390	*A03*	*12/25/2011*	*A02**	*10/25/2011*	
OptiPlex 745	2.6.6	02/07/2012	2.6.6	02/07/2012	**P**
OptiPlex 755	A20	10/26/2011	A20	10/26/2011	**P**
OptiPlex 760	A13	11/02/2011	A13	11/02/2011	**P**
OptiPlex 780	A10	10/27/2011	A10	10/27/2011	
OptiPlex 790	*A06*	*09/13/2011*	*A05**	*06/22/2011*	
OptiPlex 960	A13	11/12/2011	A13	11/12/2011	**F1,F4,F6**
OptiPlex 980	A09	10/16/2011	A09	10/16/2011	
OptiPlex 990	*A06*	*09/14/2011*	*A05**	*06/22/2011*	**P**
Latitude D531	A11	01/01/2012	A11	01/01/2012	
Latitude D630	A18	01/06/2012	A18	01/06/2012	**F7**
Latitude D631	A11	12/29/2011	A11	12/29/2011	
Latitude E4200	A21	10/17/2011	A21	10/17/2011	
Latitude E4300	A23	10/17/2011	A23	10/17/2011	**P**
Latitude E4310	*A09*	*04/12/2012*	*A08**	*10/18/2011*	
Latitude E5400	A17	10/17/2011	A17	10/17/2011	
Latitude E5410	*A11*	*09/27/2011*	*A10**	*06/02/2011*	
Latitude E5420	*A03*	*10/18/2011*	*A02**	*07/28/2011*	
Latitude E5500	A17	10/17/2011	A17	10/17/2011	
Latitude E5510	*A11*	*09/28/2011*	*A10**	*06/02/2011*	
Latitude E5520	*A03*	*10/18/2011*	*A02**	*07/28/2011*	
Latitude E6220	*A04*	*12/14/2011*	*A02**	*09/08/2011*	
Latitude E6320	*A06*	*08/30/2011*	*A05**	*07/01/2011*	**F2**
Latitude E6400 / E6400 ATG	A30	10/17/2011	A30	10/17/2011	
Latitude E6400 XFR	?	?	A31	01/11/2012	
Latitude E6410 / E6410 ATG	*A10*	*10/03/2011*	*A09**	*06/02/2011*	
Latitude E6420 / E6420 ATG	*A06*	*08/30/2011*	*A05**	*07/01/2011*	**F2**
Latitude E6420 XFR	*A03*	*08/30/2011*	*A02**	*06/13/2011*	
Latitude E6500	A26	10/17/2011	A26	10/17/2011	**P**
Latitude E6510	*A11*	*04/12/2012*	*A10**	*10/18/2011*	
Latitude E6520	*A06*	*08/30/2011*	*A05**	*07/01/2011*	**F2**
Latitude XT3	*A02**	*12/14/2011*	*A01*	*10/16/2011*	
Latitude Z	A09	12/22/2011	A09	12/22/2011	
Latitude 13	A02	10/17/2011	A02	10/17/2011	
Precision M2300	A10	01/01/2012	A10	01/01/2012	**F7**
Precision M2400	A25	10/17/2011	A25	10/17/2011	
Precision M4300	A16	01/19/2012	A16	01/19/2012	
Precision M4400	A26	10/17/2011	A26	10/17/2011	**P**
Precision M4500	*A09*	*10/20/2011*	*A08**	*06/02/2011*	
Precision M4600	*A07*	*10/11/2011*	*A03**	*06/21/2011*	
Precision M6300	A14	01/06/2012	A14	01/06/2012	**F7**
Precision M6400	A11	10/11/2011	A11	10/11/2011	
Precision M6500	A07	10/17/2011	A07	10/17/2011	
Precision M6600	*A06*	*10/11/2011*	*A03**	*06/07/2011*	
Precision T1600	*A05*	*09/13/2011 11AM*	*A04**	*09/13/2011 10AM*	
Precision T3400	A13	12/01/2011	A13	12/01/2011	**P**
Precision T5400	A10	01/17/2012	A10	01/17/2012	
Precision T7400	A10	01/17/2012	A10	01/17/2012	**P**
Precision R5400	A09	09/21/2011	A09	09/21/2011	
Precision R5500	A03	09/21/2011	A03	09/21/2011	

** Denotes that BIOS versions after this version may have a hard dependency on this specific version. If a version newer than this version is not already installed, then this version may need to be installed first before installing later BIOS versions.*

Table 1: Dell model and signed BIOS information

The above BIOS information was found by searching the Dell support web site for certain phrases. To find the BIOS version that adds the Signed Firmware Update feature, search for:

- "*Added support for Signed Firmware BIOS*"
- "*Added support of Signed Firmware BIOS*"
- "*Added DA token to enabled Signed FW update*"
- "*BIOS flash update enhancement."*
- "*Added support for Signed Firmware Updates."*

To find the first BIOS version that is a signed BIOS version, search for:

- "*Added Signed Firmware Update Transition BIOS*"
- "*Added support for Signed Firmware Update Transition BIOS*"
- "*Added Signed Flash Updated Improvement: Signed Firmware Update Transition BIOS*" (sic)
- "*Add Signed Firmware Update feature*"
- "*Add Digital Sign Feature, Signed Firmware update*." – This version added the feature and is also the first signed BIOS version.
- "*Implement Digital Signing and DA Token for Digital Signing enablement*" – This version added the feature and is also the first signed BIOS version.
- "*Add digital sign feature*."
- "*Add digital sign*"
- "*Set KEY_TRANSITION_RELEASE to "0" for digital sign*"
- "*Note:(digital sign)*"

In order to find all relevant BIOS information, a user may need to view search results that include duplicate search results. Table 1 is not a definitive list and may contain errors so please contact the vendor for conclusive information on whether a model implements signed BIOS.

3.1.1 Testing Methodology

In order to ensure there will be no unexpected deployment problems, administrators should test their Dell models to see if CCTK can detect and enable the Signed Firmware Update feature correctly. A general strategy is outlined below. Appendix A contains a script to help with this process.

1. Install the model's BIOS version listed in the Feature Added column of Table 1 and reboot the system. Reboot the system again but this time enter the BIOS setup dialog by pressing F2 or F12 when the Dell logo displays during boot. Confirm that the Signed Firmware Update entry exists under the Security section. When the Signed Firmware Update entry is selected, confirm that the wording appears correctly on the right-hand side of the screen as described in the Manually Detecting and Enabling the Signed Firmware Update Feature section.
    - **Pass**: The entry exists under the Security section and the wording on the right-hand side is correct when viewing the entry.
    - **Fail (F1)**: The entry does not appear in the Security section or the wording on the right-hand side has unexpected text.

2. Run the script supplied in Appendix A in survey mode. If the script does not report that the feature is supported, then this is a problem that should be investigated. Try installing the next BIOS version for that model and then run the script again.
    - **Pass**: The feature is detected with istokenactive and the BIOS version really is the first version that added the feature.
    - **Fail (F2)**: The feature is detected with istokenactive but the BIOS version is *not* the first version that added the feature. In this case CCTK may detect the feature in a later version of the BIOS.
    - **Fail (F3)**: The feature is never detected with istokenactive across multiple BIOS versions that support the feature.
    - **Fail (F4)**: Detecting the feature with the CCTK istokenactive command causes reliability problems on the system.
3. Now that the feature has been manually confirmed that it exists in the current BIOS version and that the feature can be detected by CCTK, run the script from Appendix A in enable mode. This will run the CCTK token command, as noted in the Automatically Detecting and Enabling the Signed Firmware Update Feature section, as long as CCTK can detect the Signed Firmware Update feature. If the feature can't be enabled, then this is a problem that should be investigated. If a BIOS password is currently set, then the script should be modified to use the CCTK valsetuppwd option as noted in the Automatically Detecting and Enabling the Signed Firmware Update Feature section.
    - **Pass**: The feature is successfully enabled.
    - **Fail (F5)**: The feature is not successfully enabled.
    - **Fail (F6)**: Enabling the feature with the CCTK token command causes reliability problems on the system.
4. Once the feature is enabled, run the script in survey mode again and make sure the script reports the feature is supported and activated. If the script reports that the feature is either not supported or not activated, then this is a problem that should be investigated.
    - **Pass**: The feature is reported as detected and enabled by the script.
    - **Fail (F7)**: The feature is reported as not detected or not enabled by the script.
5. Reboot the system and enter the BIOS setup dialog, by pressing F2 or F12 when the Dell logo displays during boot, to ensure the feature has been enabled. The text "Signed Firmware Updated Enabled" should display in the BIOS setup dialog in the System Info, System Information, or System Board section below the BIOS version, service tag, and express service code information.
    - **Pass**: The Signed Firmware Update entry no longer exists under the Security section and the correct text exists in the System Info, System Information, or System Board section.
    - **Fail (F8)**: The Signed Firmware Update entry still exists or the correct text does not exist in the System Info, System Information, or System Board section.
6. *Optionally* install the model's BIOS version that is listed in the First Signed column of Table 1. In some cases this will result in installing the same BIOS version again since sometimes the feature was added in the same version that was the first signed version.

- **Pass**: The signed BIOS version is successfully installed.
- **Fail (F9)**: The signed BIOS version is not successfully installed.

7. *Optionally* install an older unsigned BIOS version which is typically the version before the feature was added. The BIOS update should fail when the system reboots to perform the flash operation since it will be attempting to install a previously unsigned BIOS version.
    - **Pass**: The unsigned BIOS version installation fails.
    - **Fail (F10)**: The unsigned BIOS version installation succeeds.

Critical failures are F4, F5, F6, F9, and F10. Some failures, such as F2, F3, and F7, may complicate automation through scripts but are not critical failures. Only critical failures would prevent an organization from deploying signed BIOSes to a particular model through automation.

3.1.2 Known Issues

There a number of known issues on a few models that have been tested so far. Dell is actively working on fixing the known issues mentioned in this section. If a model isn't mentioned in this section and it has an issue, then please report this to Dell so that it can be fixed.

Latitude E6520, E6420, and E6320 added the Signed Firmware Update feature to BIOS version A05. While the feature has been manually confirmed to be present in the A05 BIOS version, the CCTK istokenactive command does not detect the feature and the CCTK token command can't enable the feature. It isn't until BIOS version A06 that CCTK can detect and enable the feature. This is not a critical failure.

On the Latitude D630, the CCTK istokenactive command is not able to detect that the feature is enabled. CCTK can enable the Signed Firmware Update feature successfully though. This same issue also occurs on the Precision M2300 and the Precision M6300. This is not a critical failure.

On the OptiPlex 960, the CCTK istokenactive and token commands freeze the operating system. The only way to recover is by cycling the power. The issue was present starting with BIOS version A13. This issue has been fixed as of BIOS version A17 published on 10/24/2012. Previously the only way to enable the Signed Firmware Update feature on the OptiPlex 960 was by physically enabling the feature in the BIOS setup dialog.

3.2 HP

HP supports signed BIOSes on recent rp series, dc series, Pro series, Elite series, EliteBook series, ProBook series, Mini series, and Z series models. Current HP systems ship with signed BIOSes installed and enabled by default. Previous generations of these HP models predated NIST SP 800-147 and did not ship with signed BIOSes installed and enabled by default. Some of these models support NIST SP 800-147 starting with a specific BIOS version. See Table 2 for specific models.

The Test Result column's value is:

- **'P'** when the signed BIOS was successfully installed.

- **'F'** when the signed BIOS was not successfully installed. The number after the letter corresponds to a test case in the Testing Methodology section.
- empty when a test has not been performed on that model yet.

See the Testing Methodology section for information about the testing procedures.

Model	First Signed	Release Date	SoftPaq	Test Result
rp3000	1.07 Rev A	10/28/2011	SP54367	
rp5700	1.20 Rev A	10/03/2011	SP54250	
rp5800	2.06 Rev A	07/25/2011	SP54044	
dc5700	2.09 Rev A	12/07/2011	SP54787	
dc5750	2.36 Rev A	12/12/2011	SP54788	
dc5800	1.59 Rev A	10/11/2011	SP53609	
dc5850	3.14 Rev A	01/18/2012	SP55415	
dc7700, dx7300	1.16 Rev A	10/11/2011	SP54368	
dc7700p	3.07 Rev A	09/16/2011	SP54656	
dc7800	1.32 Rev A	08/15/2011	SP53611	
dc7900	1.26 Rev A	08/15/2011	SP54033	**P**
4000 Pro	2.02 Rev A	09/01/2011	SP54252	
6000 Pro All-in-One	1.07 Rev A	09/01/2011	SP54251	
6000 Pro, 6080 Pro MT, ms6000	2.02 Rev A	08/15/2011	SP53698	**P**
6005 Pro, ms6005	1.15 Rev A	09/01/2011	SP53769	
6005 Pro USDT	1.06 Rev A	10/03/2011	SP54255	
6200 Pro, ms6200 Pro	2.06 Rev A	06/15/2011	SP53566	
8000 Elite, 8080 Elite	1.13 Rev A	08/15/2011	SP54008	
8000f Elite	1.06 Rev A	10/28/2011	SP54786	
8100 Elite, 8180 Elite	1.13 Rev A	07/14/2011	SP53945	
8200 Elite, ms6200 Pro, 6200 Pro	2.06 Rev A	06/15/2011	SP53566	**P**
EliteBook 2540p	F.20	09/07/2011	SP54580	
EliteBook 2560p	F.21	10/24/2011	SP54884	
EliteBook 2730p	F.20	12/08/2011	SP55641	
EliteBook 2740p	F.20	09/02/2011	SP54590	
EliteBook 2760p	F.20	10/24/2011	SP55060	
EliteBook 8440p, 8440w	F.20	09/06/2011	SP54571	
EliteBook 8460p, 8460w, 8560p	F.20	09/29/2011	SP54772	
EliteBook 8540p, 8540w	F.20	09/07/2011	SP54599	
EliteBook 8560w	F.20	10/24/2011	SP55072	
EliteBook 8740w	F.20	09/05/2011	SP54598	
EliteBook 8760w	F.21	10/24/2011	SP54885	
ProBook 4230s, 4330s, 4331s, 4430s, 4431s, 4530s, 4730s	F.20	10/11/2011	SP54862	
ProBook 4325s, 4326s, 4425s	F.20	09/15/2011	SP54689	
ProBook 4320s, 4321s, 4420s, 4421s	F.20	09/09/2011	SP54636	
ProBook 4410s, 4411s, 4510s, 4710s	F.20	12/09/2011	SP55876	
ProBook 4520s, 4720s	F.20	09/08/2011	SP54639	
ProBook 4525s	F.20	09/15/2011	SP54687	
ProBook 4535s, 4436s, 4435s	F.20	11/11/2011	SP55417	
ProBook 5220m	F.20	09/05/2011	SP54594	
ProBook 5310m	F.20	12/14/2011	SP55643	
ProBook 5320m	F.20	09/16/2011	SP54685	
ProBook 5330m	F.20	10/20/2011	SP55059	
ProBook 6360b, 6460b, 6560b	F.20	09/20/2011	SP54772	
ProBook 6450b, 6550b	F.20	09/02/2011	SP54664	
ProBook 6440b, 6540b	F.20	09/06/2011	SP54642	
ProBook 6445b, 6545b	F.20	11/22/2011	SP55455	

Model	First Signed	Release Date	SoftPaq	Test Result
ProBook 6445b, 6545b	F.20	11/22/2011	SP55884	
ProBook 6455b, 6555b	F.20	09/15/2011	SP54675	
ProBook 6465b, 6565b	F.20	11/14/2011	SP55307	
Mini 5102	F.20	09/09/2011	SP54640	
Mini 5103	F.20	09/09/2011	SP54637	
320, 321, 420, 421, 620, 621 Notebook PC	F.20	12/12/2011	SP55878	
325, 326, 425 625 Notebook PC	F.20	09/15/2011	SP54676	
515, 516, 615 Notebook PC	F.20	12/12/2011	SP55877	
2210b Notebook PC	F.20	02/12/2011	SP55557	
2230s Notebook PC	F.20	12/09/2011	SP55873	
2510p Notebook PC	F.20	12/07/2011	SP55558	
2710p Notebook PC	F.20	12/08/2011	SP55572	
6515b, 6715b, 6715s Notebook PC	F.20	12/02/2011	SP55556	
6510b, 6710b, 6710s Notebook PC	F.20	12/01/2011	SP55553	
6530b, 6730b Notebook PC	F.20	12/08/2011	SP55644	
6530s, 6531s, 6730s, 6830s Notebook PC	F.20	12/09/2011	SP55875	
6535s, 6735s Notebook PC	F.20	12/09/2011	SP55874	
8510p, 8510W Notebook PC	F.20	12/02/2011	SP55555	
8710p, 8710w Notebook PC	F.20	12/07/2011	SP55551	
4320t Mobile Thin Client	F.20	09/07/2011	SP54638	
Z200 Workstation	1.12 Rev A	09/23/2011	SP54673	
Z210 Workstation	1.20 Rev A	09/26/2011	SP54706	
Z400 Workstation	03.54 Rev A	11/21/2011	SP55384	
Z600 Workstation	03.54 Rev A	11/21/2011	SP55385	
Z800 Workstation	03.54 Rev A	11/21/2011	SP55386	

Table 2: HP model and signed BIOS information

The above BIOS information was found by searching the HP web site for certain phrases. To find the first signed BIOS version for an HP model, search for:

- "*Provides improved security for the BIOS flashing process*"
- "*After installing this BIOS version onto the system, prior BIOS versions cannot be installed onto the system*"
- "*After installing this BIOS version, prior versions of the BIOS cannot be reinstalled*"
- "*Improves security of the BIOS flashing process under certain circumstances by adding a stronger flash process verification algorithm*"
- "*Improves the security of the BIOS flashing process*"
- "*After this BIOS update has been installed, previous BIOS versions cannot be installed*."
- "*After this BIOS update has been installed, prior BIOS versions cannot be installed*."
- "*After installing this BIOS version onto the system, unsigned BIOS versions cannot be installed onto the system*."

In order to find all relevant BIOS information, a user may need to view search results that include duplicate search results. Table 2 is not a definitive list and may contain errors so please contact the vendor for conclusive information on whether a model implements signed BIOS.

3.2.1 Testing Methodology

In order to ensure there will be no unexpected deployment problems, administrators should test their HP models to see if the BIOS behaves as expected. A general strategy is outlined below.

1. Install a BIOS version that is equal to or newer than the version in the First Signed column of Table 2 and reboot the system.
    - **Pass**: The system boots correctly and reports the correct BIOS version.
    - **Fail (F1)**: The system does not boot correctly or does not report the correct BIOS version.
2. *Optionally* install an older unsigned BIOS version which is typically the version before the feature was added. The BIOS update should fail when the system reboots to perform the flash operation since it will be attempting to install a previously unsigned BIOS version.
    - **Pass**: The unsigned BIOS version installation fails.
    - **Fail (F2)**: The unsigned BIOS version installation succeeds.

4 Update Procedure

In an effort to provide the simplest, universal, and most reusable solution for deploying BIOS updates, a Windows batch file script could be used. This method assumes a number of requirements about the network environment:

1. The ability to push files to a client. The files would include an update script, the BIOS update, and an encrypted password file if a password is specified for HP.
2. The ability to execute the above files and executables at an Administrator or higher (SYSTEM) privilege level on the client.
3. The ability for the script to inspect the client operating system for various pieces of information, such as manufacturer, model, and BIOS version, via built-in operating systems tools during script execution.
4. The ability to install BIOS configuration tools on the client and execute them from the script.

For items 1 and 2, Active Directory could be one means of deploying and executing BIOS updates. Specifically this could be done via Group Policy Computer Startup and Shutdown scripts since they execute with appropriate privileges and the scripts are copied to the local system. The script could also copy any necessary resources from the server to the client as long as the network shares are configured correctly[21]. Almost any common systems management software should be capable of meeting these requirements.

For item 3, the built-in Windows command line tool wmic.exe can be used to retrieve the system manufacturer, system model, and BIOS version information. This information can then be used in the script to make sure it runs on the correct systems and to determine if the BIOS needs to be updated.

[21] See the 'Initial setup and configuration' and 'How to use a Group Policy-based computer script' startup sections from http://support.microsoft.com/kb/891716 for more information on how to configure shares.

For item 4, the appropriate BIOS configuration tool should be installed on the system before any BIOS updates are done. This guide only discusses BIOS management tools that are free for commercial use and that are officially supported by the vendor.

For Dell systems, the Client Configuration Toolkit (CCTK) can be used to detect and enable the Signed Firmware Update feature and, configure other BIOS settings, and manage BIOS passwords. For HP systems, SoftPaq files that contain BIOS updates also include the HPQFlash tool which is the HP BIOS update executable. These SoftPaq files also include the HPQPwd tool which is the tool that generates the encrypted password file used by HPQFlash. An administrator can also use HP's System Software Manager (HP SSM) which includes a tool called BiosConfigUtility that can be used to configure other BIOS settings and manage BIOS passwords.

4.1 Dell

Updating a Dell system to use the signed BIOS update mechanism involves updating the BIOS version to a version that includes the Signed Firmware Update feature. Next update to the latest BIOS version which will be a signed version. Then enable the Signed Firmware Update feature.

4.1.1 Preparation

To prepare for a deployment, an administrator will need to download the following items:

1. The hard dependency BIOS version, if the model has one, for each model.
2. The latest BIOS version for each model.
3. The latest version of CCTK.

After downloading a BIOS update executable, compare the MD5 or SHA1 hash published on the BIOS download page against the hash of the downloaded executable. A corrupted BIOS update executable will cause problems during deployment.

Some Dell BIOS versions have a hard dependency on a BIOS version that must be installed first. Before downloading a BIOS update, check the Important Information section, if it exists, on the BIOS download page to see if it mentions that a specific BIOS version must be installed first. Not all BIOS updates have a hard dependency on a previous version. The BIOS versions that are a hard dependency are noted with an asterisk in Table 1. Download the latest BIOS version for the models an administrator will need to update. If a model has a hard dependency on a specific BIOS version, then also download that version.

Download the latest version of CCTK to manage the Signed Firmware Update feature. If a BIOS password is set, then use CCTK to manage BIOS passwords. An administrator or script will have to clear the existing BIOS password before installing the BIOS update. Don't forget to set the BIOS password back after completing the update. Deploy CCTK to the Dell clients on the network using the organization's preferred software deployment mechanism before installing BIOS updates.

4.1.2 Process

Below is the general logic for the update process.

1. If the model has a hard dependency on a specific BIOS version and the currently installed BIOS version is less than the hard dependency version
    1. Clear the BIOS password if it exists.
    2. Execute the update in silent mode.
    3. Reboot.
    4. Set the BIOS password back if one was previously set.
2. If the currently installed BIOS version is less than the latest available version
    1. Clear the BIOS password if it exists.
    2. Execute the update in silent mode.
    3. Reboot.
    4. Set the BIOS password back if one was previously set.
3. If the Signed Firmware Update feature exists and it is not enabled, then enable the feature. No reboot is required. If a BIOS password is set, then use the CCTK valsetuppwd option, as mentioned in the Automatically Detecting and Enabling the Signed Firmware Update Feature section, when enabling the feature.

Enabling the Signed Firmware Update feature could have been done after installing the Featured Added or hard dependency BIOS version (post reboot). There have been cases where CCTK has been unable to detect the feature correctly on some models' Feature Added version as noted in the Known Issues section. In order to overcome this issue in a uniform way for an enterprise deployment, it is easier to enable the Signed Firmware Update feature after installing the latest available BIOS version for all models rather than manually testing each specific Dell model and its various BIOS versions for this issue. See Appendix B for an example script that contains the logic to accomplish the high level tasks outlined in the steps above.

Only set the BIOS password back after rebooting from the BIOS update. This is due to some Dell models that have an issue where the BIOS update will fail when the BIOS password is set back before rebooting. One model that has this problem is the OptiPlex 760.

4.1.3 Commands

Dell generally has two different types of BIOS update packages. Older Dell models use a BIOS update package with one set of command line options while newer Dell models use a different set of command line options. One way to find out if a model uses a newer or older BIOS update package is by running the update executable from the command line with the /? option. If help text or help dialog is displayed, then it is a newer BIOS update package. If an error message about an invalid command line is displayed, then it is an older BIOS update package. The newer update packages are actually called Dell Update Packages (DUP)[22]. DUPs have different command line options and error codes than older BIOS update packages.

[22] Dell Update Package documentation. http://support.dell.com/support/edocs/software/smdup

4.1.3.1 Non-DUP BIOS Updates

Non-DUP executables are BIOS update packages for older Dell systems. The supported command line options[23] are controlled by the BIOS vendor. Example commands are below.

- **O755-A20.exe /nopause** – Runs the BIOS update in silent mode. A reboot will happen.
- **O755-A20.exe /nopause /noreboot** – Runs the BIOS update in silent mode. A reboot will NOT happen.
- **D630-A18.exe /nopause /noreboot /forceit** – Forcefully runs the BIOS update in silent mode. A reboot will NOT happen. Laptops may need the forceit option if they are not plugged in or if the battery is old and "burned out". Even if a laptop is plugged in and the battery is "burned out", the forceit option may still be needed.

In a batch script, the %errorlevel% variable should be 0 when an update executes successfully. Any non-zero value should be treated as a failure. They are not documented by Dell.

Old BIOS update packages do not have a command line option to specify a BIOS password. CCTK must be used to clear the BIOS password before the update and then used again to set it back after rebooting from a BIOS update.

- **cctk.exe --setuppwd= --valsetuppwd=*currentpassword*** – Clears the password. The error level should be 0.
- **cctk.exe --setuppwd=*password*** – Sets the password. The error level should be 0.

Dell's tools do not support providing an encrypted password unlike other vendors such as HP and Fujitsu.

CCTK error codes are documented in the official CCTK documentation and on the Dell TechCenter page for CCTK error codes[24]. There are specific error codes related to passwords and they may be relevant to scripts or any automation process that an administrator develops.

4.1.3.2 DUP BIOS Updates

DUP BIOS executables have a standard set of supported command line options[25] that Dell controls. Example commands are below.

- **O990-A05.exe /s** – Runs the BIOS update in silent mode. A reboot will NOT happen.
- **O990-A05.exe /s /r** – Runs the BIOS update in silent mode. A reboot will happen immediately.
- **O990-A05.exe /s /r /f** – Forcefully runs the BIOS update in silent mode. A reboot will happen immediately.

[23] Legacy BIOS Updates. http://en.community.dell.com/techcenter/enterprise-client/w/wiki/3461.legacy-bios-updates.aspx
[24] Dell Client Configuration Toolkit Error Codes. http://en.community.dell.com/techcenter/systems-management/w/wiki/1953.dell-client-configuration-toolkit-cctk-error-codes.aspx
[25] DUP BIOS Updates. http://en.community.dell.com/techcenter/enterprise-client/w/wiki/3462.dup-bios-updates.aspx

- **O990-A05.exe /s /r /p=*password*** –Runs the BIOS update in silent mode and specifies a BIOS password. A reboot will happen immediately.
- **O990-A05.exe /s /l="C:\BIOS update logs\O990-A05log.txt"** – Runs the BIOS update in silent mode and logs information to the specified log file. A reboot will NOT happen. This is useful for troubleshooting when a BIOS update does not execute successfully.

In a batch script, the %errorlevel% variable should be 2 (reboot required) when just using the /s option or 6 (rebooting system) when using the /s and /r options. As mentioned before, DUP BIOS executables have a /? option that explains the supported options. DUP documentation[26] is also available on Dell's support web site. The DUP User's Guide documents the command line options and most error codes. A list of error codes and their meaning are included in Table 3.

Code	Meaning	Description
-1	Cancel	User manually canceled the update when using the GUI.
0	Successful	This value isn't returned for BIOS updates because they always require a reboot. 0 would be returned from a DUP that doesn't require a reboot.
1	General failure	There are probably many cases when this value is returned. One case when this value is returned is when the hard dependency BIOS version was not already installed on the system. Another common case is when incompatible command line options are used together.
2	Reboot required	This value is returned when a successful update happens and just the /s option is used.
3	Soft dependency failure	One example when this value is returned is when a system is updated to the same BIOS version that is already installed or a system is rolled back to an older BIOS version. This failure can be overridden with the /f option.
4	Hard dependency failure	This value is returned when some required dependency was not met. Try running the update again with the log option (/l="C:\mylog.txt") to get more detailed failure information. This failure *cannot* be overridden with the /f option.
5	Hard error	The update doesn't apply to the system (OS not supported, model not supported, etc). This failure *cannot* be overridden with the /f option.
6	Rebooting system	This value is returned when a successful update happens and both the /s and /r options are used.
7	Invalid password or password validation failure	This value is returned when there is a password error. If an administrator uses the /s option and forgets to supply a password, then a password prompt will *not* be displayed. If an administrator supplies the wrong password, then a prompt is displayed though.
8	Unknown	
9	RPM verification failed	This value does not apply to Windows operating systems.

Table 3: List of DUP error codes and meanings

The /f option should be avoided on DUP BIOS executables because it suppresses some real error codes. For example, if an administrator or script used the /f option without installing the hard dependency BIOS version on the system first, then the latest BIOS version will still return a value of 2 (reboot required) which is misleading. The logging option (/l="C:\mylog.txt") can be used to generate a log file that will have more details about the BIOS update failure.

Once the system has rebooted, a script can use **cctk.exe --completioncode** to check if the BIOS update was successful. The command should return 0000 if the update was completed. It can also return FFFF on systems where no BIOS update has been attempted. All other values returned by this command should be considered an error which means the BIOS update was *not* successful. The completion code of

[26] Dell Update Package documentation. http://support.dell.com/support/edocs/software/smdup

0001 is the value used to denote when an unsigned BIOS update was attempted on a system that has the Signed Firmware Update feature enabled. A list of BIOS completion codes and their meanings are documented on the Dell TechCenter web site[27].

Unfortunately some models do not appear to update the completion code value correctly. Some models may update the completion code to 0001 if the Intel Management Engine option ROM update failed but the main BIOS update succeeded. Some models appear to never set the value to anything other than FFFF despite a successful or failed BIOS update. Administrators should run the script from Appendix A in survey mode after performing the test in step 7 from the Testing Methodology section to see if the correct value is returned by the CCTK completioncode option for the different models they will update.

4.2 HP

HP models are fairly straightforward to update to a signed BIOS. Installing HP signed BIOS updates are just like installing unsigned BIOS updates. HP has no special BIOS feature that needs to be enabled to enforce signed BIOS updates which means installing a BIOS configuration tool is optional.

4.2.1 Preparation

To prepare for a deployment, an administrator will need to download the following items:

1. The latest BIOS version for each model.
2. The latest version of HP SSM.

Once an administrator downloads the BIOS update SoftPaq from the specific HP model page, extract the SoftPaq. Generally it extracts to C:\swsetup\sp###### where # is the SoftPaq number. An administrator can use the -e option (-e"C:\path") to specify a different path to extract the contents to. Once the contents are extracted, navigate into the numbered SoftPaq folder. In that folder an administrator will find various files and folders but the most important one is the HPQFlash folder. This folder contains HPQFlash.exe which is the BIOS update executable. It also contains ROM.cab which is the actual BIOS update file. By default HPQFlash looks for the existence of ROM.cab and will use it if it is found when HPQFlash is executed. Rename the cab file to denote the specific model and BIOS revision such as dc7900-126RevA.cab. Then use HPQFlash's -f option and specify the path to the cab file in order to make sure the system is updated to the correct version when using a script to update many different models.

If a BIOS password is set, then an administrator needs to decide which BIOS management tool they want to use. An administrator needs to decide if they want to use HPQFlash with an encrypted password that is generated by HPQPwd or if they want to use the HP SSM BiosConfigUtility which only supports plaintext passwords.

Even if an administrator doesn't use HP SSM to manage BIOS passwords, they may still find it beneficial to install it on their systems since they can configure any BIOS option using the REPSET functionality and the setconfig option.

[27] Dell Client Configuration Toolkit BIOS Completion Codes. http://en.community.dell.com/techcenter/systems-management/w/wiki/3383.aspx

HP SSM is available for free from HP's web site[28] or by directly downloading it from HP's FTP site[29]. The latest version is 2.15 Rev A and the SoftPaq is sp52095.exe. It can be silently installed by running **sp52095.exe /s** and it will extract to the %ProgramFiles%\Hewlett-Packard\SSM folder by default. The SSM folder will contain BiosConfigUtility.exe that can be used to set, clear, and change BIOS passwords. The password will need to be specified in plaintext though because the utility does not support using encrypted password files generated by HPQPwd.

Running **sp52095.exe /s /f"C:\path\to\install\"** will install the tool to a different location. The SoftPaq merely extracts the data to the folder so an administrator or script only needs to delete the folder to uninstall SSM. This is true regardless if the default install location is used or a custom location is used.

4.2.2 Process

Below is the general logic for each potential update process. Using BiosConfigUtliity makes the update process slightly more complicated compared to using HPQPwd.

1. If the current BIOS version on the system is less than the latest version
    1. Run latest BIOS update in silent mode.
        - If a BIOS password is set, then provide an encrypted password file, generated with HPQPwd, to the BIOS update executable.
    2. Reboot.

or

1. If the current BIOS version on the system is less than the latest version
    1. Clear the BIOS password, if it exists, by using BiosConfigUtility.
    2. Run the latest BIOS update in silent mode.
    3. Reboot.
    4. Set the BIOS password back if there was one set previously.

So far it seems safe to set the password back on HP systems before rebooting but more testing is required to be conclusive. Obviously the first option is simpler. If a BIOS password is not set, then just run the BIOS update executable and reboot.

4.2.3 Commands

Below are some example commands to perform BIOS updates on HP systems.

- **HPQFlash.exe -s -f"C:\path\to\file.cab"** – Performs a silent update using the specified cab file. No reboot happens.
- **HPQFlash.exe -a -s -f"C:\path\to\file.cab"** – Forcefully performs a silent update using the specified cab file. No reboot happens.

[28] HP System Software Manager. http://www.hp.com/go/ssm
[29] HP SSM 2.15 Rev A. ftp://ftp.hp.com/pub/softpaq/sp52001-52500/sp52095.exe

- **HPQFlash.exe -a -s -f"C:\path\to\file.cab" -p"C:\path\to\encryptedpassword.bin"** – Forcefully performs a silent update using the specified cab file and the specified encrypted BIOS password file. No reboot happens. The encrypted password file is generated by HPQPwd.

In a batch file, HPQFlash will set the %errorlevel% variable value to 0 when successful. When there is a problem then it appears to return an errorlevel value of 259. When inspecting the log file generated by HPQFlash (HPQFlash.log will be in the same directory as HPQFlash.exe), an administrator can check the **Error Code** and **Return Code** fields in the log. Table 4 lists some error codes and return codes that have been encountered.

Return Code	Error Code	Description
0x103	**0x101**	PASSWORD_REQUIRED. A BIOS password is set but the -p option was not used to specify an encrypted password file.
0x103	**0x112**	INCORRECT_PASSWORD_FILE. A BIOS password is set but the password file was not found.
0x103	**0x113**	INCORRECT_PASSWORD. A BIOS password is set but the encrypted password was the wrong password.
0x103	**0x107**	EXPLODE_CAB_FAIL. A BIOS update cab file could not be found or was corrupted.
0x103	**0x000000f0**	An attempt was made to roll back to an unsigned BIOS version. This could possibly have other meanings though.
0x642	**0x111**	SAME_OR_OLD_BIOS. The BIOS update did not apply because the update was the same, or older than, the currently installed BIOS version. Use the -a option with HPQFlash to force the update.

Table 4: HPQFlash return codes and error codes

As mentioned before, an alternate to using HPQFlash with an encrypted password file is using BiosConfigUtility that is included with HP SSM. It can be used to manage BIOS passwords and perform pretty much any BIOS configuration using the REPSET functionality and the setconfig option.

- **BiosConfigUtility.exe /NewSetupPassword:"newpassword"** – Sets a new BIOS password.
- **BiosConfigUtility.exe /CurSetupPassword:"currentpassword" /NewSetupPassword:"newpassword"** – Changes the BIOS password.
- **BiosConfigUtility.exe /CurSetupPassword:"currentpassword" /NewSetupPassword:""** – Clears the BIOS password.
- **BiosConfigUtility.exe /ADVANCED /GETCONFIG:"C:\currentbiossettings.txt"** – Retrieves the current BIOS configuration names and values and also denotes the currently selected value for each option.
- **BiosConfigUtility.exe /SETCONFIG:"C:\desiredbiossettings.txt"** – Sets new BIOS configuration values.
- **BiosConfigUtility.exe /SETCONFIG:"C:\desiredbiossettings.txt" /CurSetupPassword:"currentpassword"** – Sets new BIOS configuration values and specifies the current BIOS password since one of the configuration options required a password to set a new value.

For consistency, always use quotes around the password since BiosConfigUtility expects empty quotes when clearing a password.

When using BiosConfigUtility in a batch script, it will return an error level of 0 when successful. All other non-zero error codes should be treated as a failure. The most common error code is 10 since this error code is used for all password related errors. The error codes noted in Table 5 are taken from the PDF file inside the SSM folder.

Error Code	Description
0	Success.
1	Not supported.
2	Unspecified error.
3	Timeout.
4	Failed.
5	Invalid parameter.
6	Access denied.
10	The specified BIOS password was wrong or missing.
11	The supplied REPSET file was empty or the REPSET file was not found when.
12	'English' was not the first line of the supplied REPSET file.
13	A BIOS setting was not changed.
14	Unable to write data to the BIOS.
15	The supplied REPSET file contained incorrect or invalid syntax.
16	This is a general catch all error code.
17	The help dialog was displayed because the help option was used or the tool was invoked incorrectly.
18	Setting is unchanged.
19	Setting is read only.
20	Invalid setting name.
21	Invalid setting value.

Table 5: BiosConfigUtility error codes

5 General Deployment Suggestions

It is common for administrators to be concerned about updating the BIOS on their systems. Many are concerned that a system could be rendered inoperable due to a BIOS update which is commonly referred to as 'bricking'. While this issue may have been more common in past decades, it is very rare for this to happen now. A customer pilot in 2011 required BIOS updates, before signed BIOSes were available, to about 2000 HP and Dell systems and all systems were updated without any bricking. Even if a system were to become inoperable, most vendors have advanced support personnel that can recover the system back to a working state.

Many administrators are unfamiliar with BIOS and have usually never updated the BIOSes on their network. Once administrators experience the BIOS update processes on their own test systems, most concerns should subside. Installing a BIOS update is very similar to installing an operating system update and shouldn't be treated much different than other updates that require a system reboot.

Below are a few tips to make the overall process as smooth as possible.

- Ensure that the system maintains power while the BIOS update is applied during the system boot.
- It is possible that a BIOS download may get corrupted when downloading using HTTP or FTP. Dell publishes MD5 and SHA1 hashes for their BIOS updates on the specific BIOS version's download page. Administrators should check the hash of their downloaded BIOS file and

compare it with the posted hashes. HP does not post hashes for an administrator to check against.

- Always install and test BIOS updates on non-production systems first. This will ensure the BIOS download has not been corrupted and works properly on the specific model. This will also allow administrators to identify and understand any model specific issues.
- Deploy BIOS updates per model. For example, update all OptiPlex 960s, and then update all OptiPlex 990s, then update all HP 6000s, etc. If a problem occurs, then they usually occur per model. Deploying the updates per model will make troubleshooting problems much easier.
- Make sure laptops are plugged into a power outlet before updating the BIOS. Some BIOS updates check that the battery is charged above a certain level but this may not be the case for all models. Some laptops may no longer detect and report the battery level correctly so the BIOS update force option, if available, may need to be used. Some models' BIOS updates do not respect the force option and may still result in a failed BIOS update when the update is applied during boot. This will not damage the system.
- Since CCTK may have issues detecting the signed BIOS feature on specific models and BIOS versions, update the BIOS to version listed in the Featured Added column in Table 1 first, then update to the latest BIOS version, and then enable the signed BIOS feature.
- BIOS passwords slightly complicate the overall update process but not in a way that makes the update process unfeasible. Both HP and Dell have tools for BIOS password management that work without any issues. Some specific models and BIOSes may have issues with passwords and BIOS updates though. It is strongly encouraged to test how setting and clearing BIOS passwords affects the automation process used to deploy BIOS updates. Model and BIOS specific issues have occurred in a few Dell systems.
- If BIOS passwords are set, then consider the security ramifications of how the scripting or automation process stores and uses the password. BIOS passwords stored in clear text on clients and servers and sent in unencrypted network traffic is a security risk. HP tools support providing encrypted password files to their BIOS update executables while Dell tools do not.
- Since a BIOS update could fail, and the update automation process should be stateless, scripts must be careful to not repeatedly attempt BIOS updates on systems where the updates fail. A maximum BIOS update failure retry mechanism may need to be implemented to prevent unlimited BIOS update attempts.
- If a system has BitLocker enabled, then upgrading its BIOS will change the Platform Configuration Register (PCR) values that BitLocker measures. This will cause BitLocker to prompt the user for the recovery password since the BIOS measurements have changed. Administrators may want to temporarily disable BitLocker on systems before upgrading the BIOS and then enable BitLocker again after the BIOS upgrade has completed.

A successful enterprise wide BIOS update process can be achieved as long as administrators thoroughly plan and test their update strategy as discussed in this guide. Vendors that support signed BIOSes

currently ship systems with NIST SP 800-147 compliant BIOSes installed. Installing signed BIOS updates on older systems is recommended when a vendor provides signed BIOSes for older systems.

6 Appendices

The appendices contain example scripts that may benefit administrators with testing, enabling, and deploying Dell signed BIOSes.

6.1 Appendix A

The **dellsignedbios.bat** script can be used to survey a Dell system to see if the current BIOS version supports the Signed Firmware Update feature, to see if the feature is currently enabled, and enable the feature if it is supported and not currently enabled. It requires Dell's CCTK software to be installed on the system. The script's options are:

- **dellsignedbios.bat** – Displays the help dialog.
- **dellsignedbios.bat /survey** – Runs a survey on the system.
- **dellsignedbios.bat /enable** – Enables the feature.
- **dellsignedbios.bat /enable /force** – Attempts to enable the feature even if CCTK can't detect the feature exists.

If a BIOS password is currently set, then the script should be modified to use the CCTK valsetuppwd option when enabling the feature as noted in the Automatically Detecting and Enabling the Signed Firmware Update Feature section.

When copying the script from the document, be careful that commands do not get wrapped onto new lines. Script syntax errors may occur if a command gets wrapped onto a new line.

```
@echo off

setlocal enabledelayedexpansion enableextensions

rem ------------------------------------------------------------------

set SIGNED_TOKEN=0x0325

set IS_SUPPORTED=false
set IS_ENABLED=false
set CCTK_BIOS_ERROR=false

set IS_64=false

set MODE=.
set FORCE=false

set ERR_MSG=.

set PROG_PATH=.
set CCTK=.
set CCTK_MSG=.
set CCTK_CODE=.

set MANUFACTURER=.
set MODEL=.
set BIOS_VER1=.
set BIOS_VER2=.
set BIOS_DATE=.
set COMP_CODE=.

rem ------------------------------------------------------------------

if %1.==. (
   echo.
```

```
   echo no option was specified
   echo.
   goto printhelp
) else (
   set MODE=%1
)

rem remove the slash
set MODE=%MODE:/=%

if /i %MODE%==survey (
   goto begin
)

if /i %MODE%==enable (
   if /i %2%.==. (
      set FORCE=false
      goto begin
   ) else (
      set ARG=%2
      set ARG=!ARG:/=!

      if /i !ARG!==force (
         set FORCE=true
         goto begin
      ) else (
         echo.
         echo '!ARG!' is an invalid sub-option for the enable option
         echo.
         goto printhelp
      )
   )
)

echo.
echo '%MODE%' is an invalid option
echo.

goto printhelp


:begin

if /i %PROCESSOR_ARCHITECTURE%==x86 (
   goto wow64test
) else (
   set IS_64=true
   goto setpaths
)


:wow64test

if not defined PROCESSOR_ARCHITEW6432 (
   set IS_64=false
) else (
   set IS_64=true
)


:setpaths

rem do NOT mess with the code below unless you understand how parens and enabledelayedexpansion
rem interact AND the fact that the variables have parens in the actual data

if /i %IS_64%==true (
   set PROG_PATH=%ProgramFiles(x86)%
   set CCTK=!PROG_PATH!^)\Dell\CCTK\X86_64\cctk.exe
   goto prepwmi
)

if /i %IS_64%==false (
   set PROG_PATH=%ProgramFiles%
   set CCTK=!PROG_PATH!\Dell\CCTK\X86\cctk.exe
   goto prepwmi
)


:prepwmi

rem prime WMI in the case MOFs need to be compiled. do this for each unique class used in the script

wmic /locale:ms_409 COMPUTERSYSTEM GET Manufacturer,Model /VALUE >nul
wmic /locale:ms_409 BIOS GET SMBIOSBIOSVersion /VALUE >nul


rem get system properties

for /f "tokens=2 delims==" %%A in ('wmic /locale:ms_409 COMPUTERSYSTEM GET Manufacturer /VALUE') do set MANUFACTURER=%%A
for /f "tokens=2 delims==" %%B in ('wmic /locale:ms_409 COMPUTERSYSTEM GET Model /VALUE') do set MODEL=%%B
for /f "tokens=2 delims==" %%C in ('wmic /locale:ms_409 BIOS GET SMBIOSBIOSVersion /VALUE') do set BIOS_VER1=%%C
```

```
rem make sure this is a Dell system

echo %MANUFACTURER% | findstr /i /c:"Dell" >nul

if not %errorlevel%==0 (
   set ERR_MSG=System was not a Dell. It was %MANUFACTURER%
   goto printerror
)

rem have to use exclamation marks inside the if statement on the variable so that it does not break
rem the script due to how parens and enabledelayedexpansion interact AND the fact that the variables
rem have parens in the actual data

if not exist "%CCTK%" (
   set ERR_MSG=CCTK did not exist at '!CCTK!'
   goto printerror
)


for /f "tokens=2 delims==" %%D in ('"%CCTK%" --biosver') do set BIOS_VER2=%%D
for /f "tokens=2 delims==" %%E in ('"%CCTK%" --lastbiosupdate') do set BIOS_DATE=%%E
for /f "tokens=2 delims==" %%F in ('"%CCTK%" --completioncode') do set COMP_CODE=%%F


echo manufacturer: '%MANUFACTURER%'
echo model: '%MODEL%'
echo.
echo bios wmic: '%BIOS_VER1%'
echo bios cctk: '%BIOS_VER2%'
echo bios date: '%BIOS_DATE%'
echo bios completion code: '%COMP_CODE%'
echo.

rem this captures the message from CCTK but the for loop overwrites errorlevel to 0 and 'hides'
rem the error level set by running CCTK so we have to run the command again after this

for /f "tokens=*" %%G in ('"%CCTK%" --istokenactive=%SIGNED_TOKEN%') do set CCTK_MSG=%%G

"%CCTK%" --istokenactive=%SIGNED_TOKEN% >nul

set CCTK_CODE=%errorlevel%

echo CCTK message: %CCTK_MSG%
echo CCTK return code: %CCTK_CODE%
echo.

if %CCTK_CODE%==0 (
   set IS_SUPPORTED=true

   echo %CCTK_MSG% | findstr /i /c:"is active" >nul

   if !errorlevel!==0 (
      set IS_ENABLED=true

      if /i %MODE%==enable (
         echo.
         echo the signed BIOS feature is already enabled
         echo.
      )

      goto printresults
   )

   echo %CCTK_MSG% | findstr /i /c:"is not active" >nul

   if !errorlevel!==0 (
      set IS_ENABLED=false

      if /i %MODE%==enable (
         goto tryenable
      )

      goto printresults
   )
) else (
   rem CCTK_CODE should be 81 if the BIOS does not support the signed BIOS feature
   rem we could also test CCTK_MSG for 'not available'
   rem CCTK sometimes does not detect the feature though so there is an option enable it anyway

   if %CCTK_CODE%==81 (
      if /i %FORCE%==true (
         goto tryenable
      ) else (
         goto printresults
      )
   ) else (
      set ERR_MSG=CCTK unexpectedly failed with error code '%CCTK_CODE%' with message '%CCTK_MSG%'
      goto printerror
   )
)
```

```
:tryenable

rem enable the feature
"%CCTK%" --token=%SIGNED_TOKEN% > nul

rem capture the return code from the token command
set CCTK_CODE=%errorlevel%

rem CCTK should report the token is active if everything is working correctly
for /f "tokens=*" %%H in ('"%CCTK%" --istokenactive=%SIGNED_TOKEN%') do set CCTK_MSG=%%H

if %CCTK_CODE%==0 (
   echo CCTK successfully enabled the signed BIOS feature
   echo.

   set IS_ENABLED=true

   rem check if istokenactive detects that the feature is enabled now

   echo %CCTK_MSG% | findstr /i /c:"is active" >nul

   if !errorlevel!==0 (
      echo CCTK correctly detected that the feature is enabled
      echo.

      set CCTK_BIOS_ERROR=false

      goto printresults
   )

   echo %CCTK_MSG% | findstr /i /c:"is not active" >nul

   if !errorlevel!==0 (
      echo CCTK did NOT correctly detect that the feature is enabled
      echo.

      set CCTK_BIOS_ERROR=true

      goto printresults
   )

   goto printresults
) else (
   set IS_ENABLED=false

   rem capture the failure message from the token command
   for /f "tokens=*" %%I in ('"%CCTK%" --token=%SIGNED_TOKEN%') do set CCTK_MSG=%%I

   set ERR_MSG=CCTK unexpected failed with error code '!CCTK_CODE!' with message '!CCTK_MSG!'

   goto printerror
)


:printresults

echo supported: %IS_SUPPORTED%
echo enabled: %IS_ENABLED%
echo cctk/bios error: %CCTK_BIOS_ERROR%

goto end

:printhelp
echo Dell signed firmware update survey and enabling script
echo 03/12/2012
echo.
echo   %0 [ /survey ^| /enable [ /force ] ]
echo.
echo   survey - checks if the signed BIOS feature exists and if it is enabled or not
echo   enable - attempts to enable the signed BIOS feature if CCTK can detect it
echo   force - force enabling the signed BIOS feature even if CCTK can not detect it
echo.
echo Examples
echo.
echo   %0 /survey
echo   %0 /enable
echo   %0 /enable /force

goto end

:printerror
echo %ERR_MSG%

:end

endlocal
```

6.2 Appendix B

Comparing Dell BIOS version numbers is generally straightforward since BIOS versions for Dell enterprise clients follow Axx (A05, A10, etc) or #.#.# (2.6.0, 2.6.4, etc) numbering formats. Replacing 'A' with nothing and '.' with nothing results in a number that can be compared using if statements and number comparison operators (EQU, LSS, LEQ, GTR, GEQ). These features can be used to handle installing the hard dependency BIOS version first, then installing the latest BIOS version second, and then finally enabling the Signed Firmware Update feature if it is present and not enabled. This script is an example of how Dell BIOS version numbers can be compared against each other.

```
@echo off

setlocal

set CUR_BIOS_VER=.
set COMP_CUR_BIOS_VER=.

set LATEST_BIOS_VER=A05
set COMP_LATEST_BIOS_VER=.

set PRE_BIOS_VER=A06
set COMP_PRE_BIOS_VER=.


for /f "tokens=2 delims==" %%A in ('wmic /locale:ms_409 BIOS GET SMBIOSBIOSVersion /VALUE') do set CUR_BIOS_VER=%%A

echo Current BIOS version: '%CUR_BIOS_VER%'
echo Prereq BIOS version: '%PRE_BIOS_VER%'
echo Latest BIOS version: '%LATEST_BIOS_VER%'
echo.


set COMP_CUR_BIOS_VER=%CUR_BIOS_VER%

rem remove periods
set COMP_CUR_BIOS_VER=%COMP_CUR_BIOS_VER:.=%

rem remove 'A' character
set COMP_CUR_BIOS_VER=%COMP_CUR_BIOS_VER:A=%

rem remove space characters
set COMP_CUR_BIOS_VER=%COMP_CUR_BIOS_VER: =%


set COMP_PRE_BIOS_VER=%PRE_BIOS_VER%

rem remove periods
set COMP_PRE_BIOS_VER=%COMP_PRE_BIOS_VER:.=%

rem remove 'A' character
set COMP_PRE_BIOS_VER=%COMP_PRE_BIOS_VER:A=%

rem remove space characters
set COMP_PRE_BIOS_VER=%COMP_PRE_BIOS_VER: =%


set COMP_LATEST_BIOS_VER=%LATEST_BIOS_VER%

rem remove periods
set COMP_LATEST_BIOS_VER=%COMP_LATEST_BIOS_VER:.=%

rem remove 'A' character
set COMP_LATEST_BIOS_VER=%COMP_LATEST_BIOS_VER:A=%

rem remove space characters
set COMP_LATEST_BIOS_VER=%COMP_LATEST_BIOS_VER: =%


echo Comparable current BIOS version: '%COMP_CUR_BIOS_VER%'
echo Comparable prereq BIOS version: '%COMP_PRE_BIOS_VER%'
echo Comparable latest BIOS version: '%COMP_LATEST_BIOS_VER%'
echo.


if %COMP_CUR_BIOS_VER% LSS %COMP_PRE_BIOS_VER% (
   echo Current BIOS version needs to be updated to pre-req BIOS version
   echo.
   goto updatebios
)


if %COMP_CUR_BIOS_VER% LSS %COMP_LATEST_BIOS_VER% (
```

```
   echo Current BIOS version needs to be updated to the latest BIOS version
   echo.
   goto updatebios
)

if %COMP_CUR_BIOS_VER% GEQ %COMP_LATEST_BIOS_VER% (
   echo BIOS needs to enable the signed updates feature if available
   echo.
   goto checkandenablefeature
)

rem should never get here unless a comparable BIOS version did not end up being a number
echo error comparing BIOS version numbers
echo.
goto end


:updatebios

echo Execute BIOS update here
echo.
goto end

:checkandenablefeature

echo Check for and enable signed update feature here if it is available
echo.
goto end


:end

endlocal
```