HANCOM
한컴MDS

Lockdown 쉽게 알기 & 팁

한컴MDS 박원철 대리

2020. 8. 19

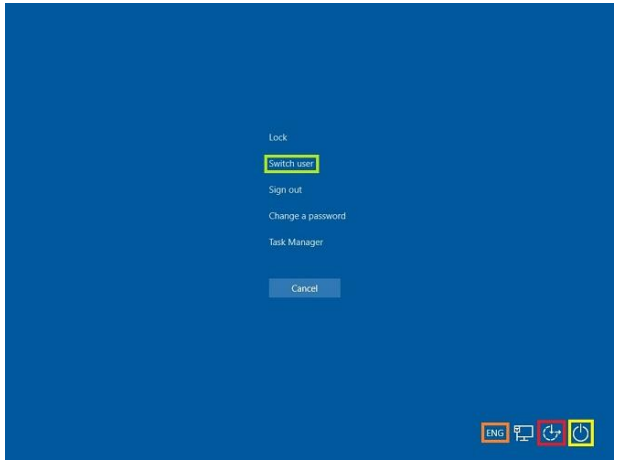
Lockdown 쉽게 알기 & 팁

I. Lockdown 기능 이란?
II. 기능 소개 & Demo
III. QnA

1. Lockdown 기능 이란?

1. Lockdown 기능 이란?

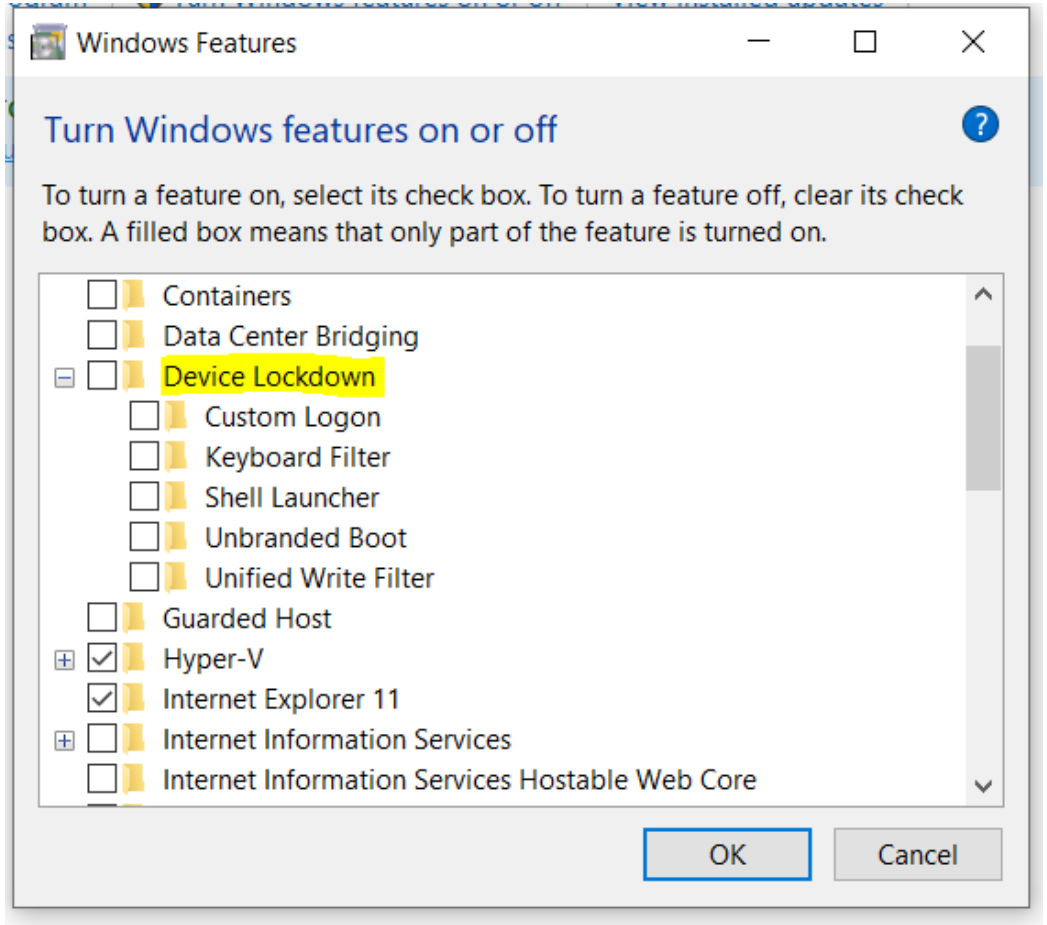
Lockdown 기능은 왜 사용해야 할까?



- 시스템 장애 방지
- 커스텀 로그온 환경
  특정 앱만 사용
- 불특정 사용자로부터의 보안
  이미지 보호

1. Lockdown 기능 이란?

Lockdown 기능은 왜 사용해야 할까?

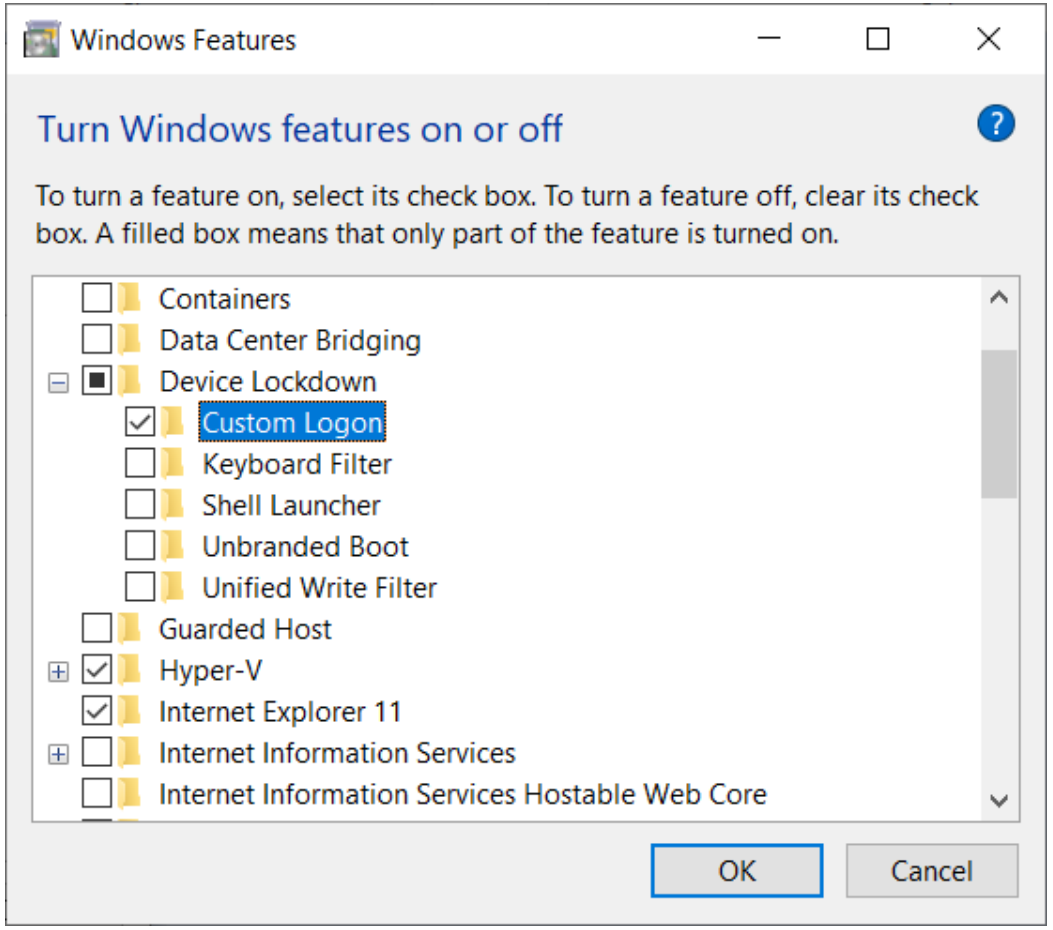


- **WES7(Windows Embedded Standard 7) EEF(Embedded Enabling Features)의 Windows 10 버전**
- **Lockdown 세부항목**
  - Custom Logon
  - Keyboard Filter
  - Shell Launcher
  - Unbranded Boot
  - Unified Write Filter

2. 기능 소개 & Demo

2. 기능 소개 & Demo

Custom Logon

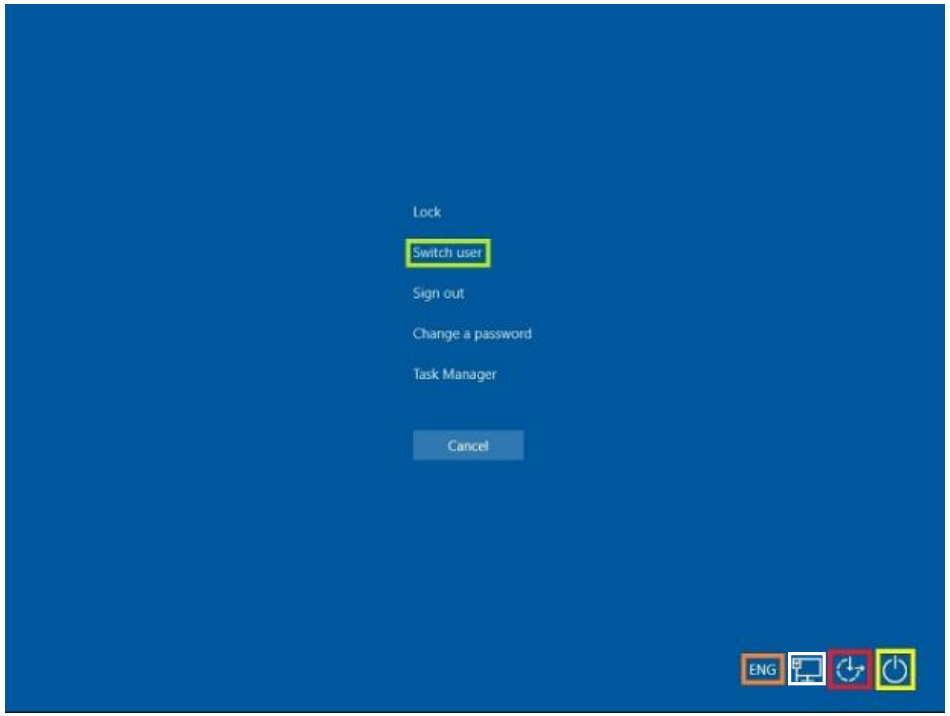


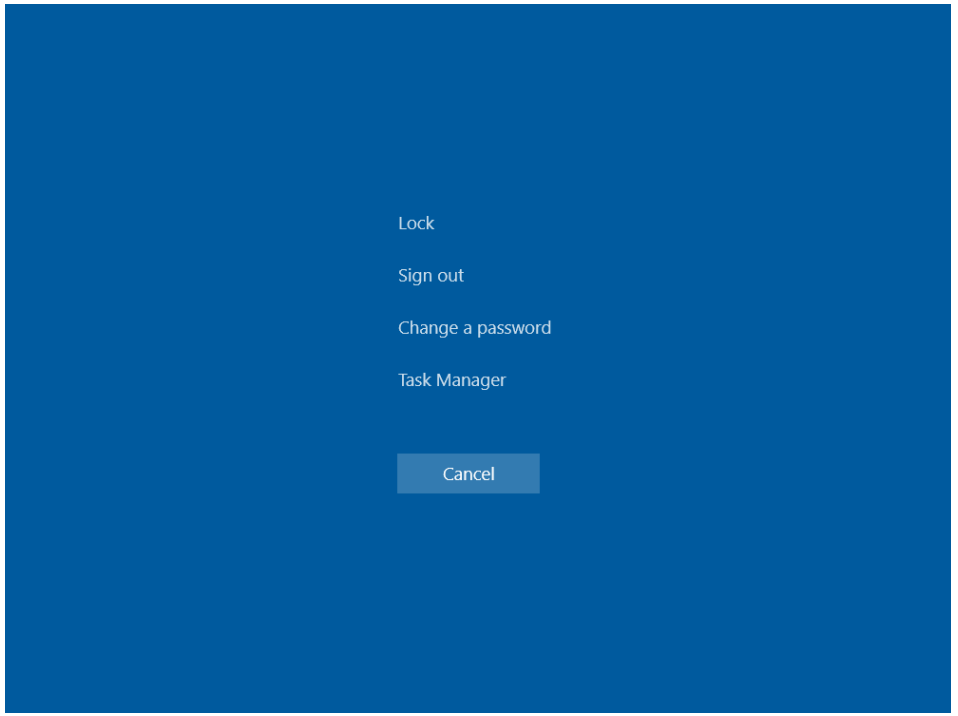
- **Welcome Screen UI 숨기기 기능 제공**
- **시스템 종료시 프로그램 종료 대기 화면 제거**
- **Windows SIM 이용한 Unattend에 Custom Logon 관련 설정 추가 가능**
- **(TMI)**
  1511, Embedded Logon
  1607 and later, Custom Logon

2. 기능 소개 & Demo

Custom Logon

- **Welcome Screen UI 숨기기 기능 제공**

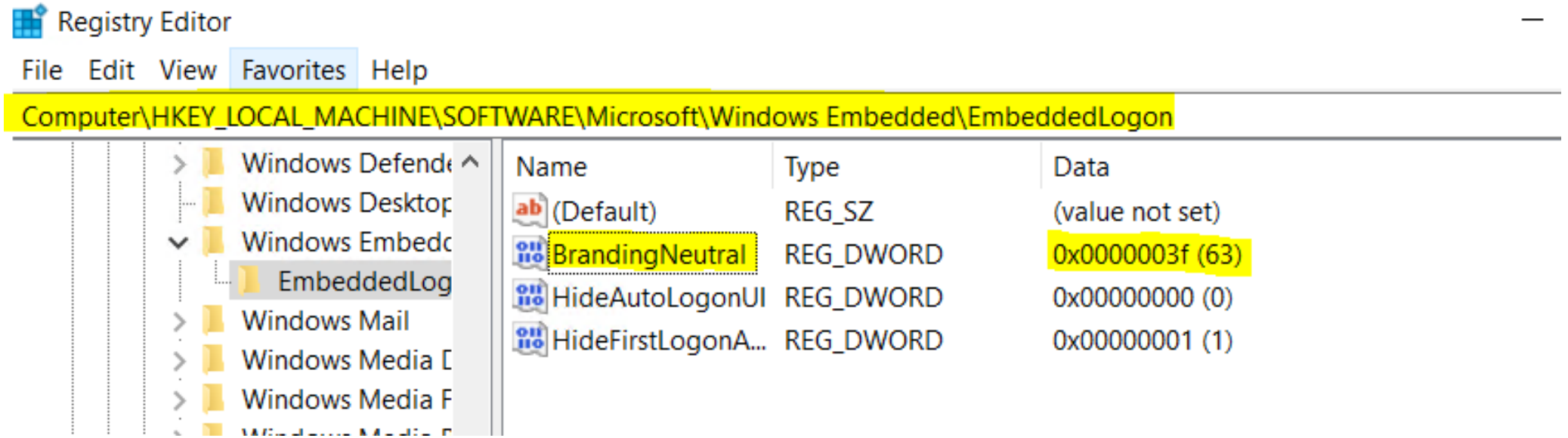




2. 기능 소개 & Demo

Custom Logon

- **Welcome Screen UI 숨기기 기능 제공**



2. 기능 소개 & Demo

Custom Logon

- **Welcome Screen UI 숨기기 기능 제공**

The following table shows the possible values. To disable multiple Welcome screen UI elements, combine these values using bitwise exclusive-or logic.

Action	Registry value
Disable all Welcome screen UI elements	**static const DWORD EMBEDDED_DISABLE_LOGON_ANCHOR_ALL = 0x1**
Disable the Power button	**static const DWORD EMBEDDED_DISABLE_LOGON_ANCHOR_SHUTDOWN = 0x2**
Disable the Language button	**static const DWORD EMBEDDED_DISABLE_LOGON_ANCHOR_LANGUAGE = 0x4**
Disable the Ease of Access button	**static const DWORD EMBEDDED_DISABLE_LOGON_ANCHOR_EASEOFACCESS = 0x8**
Disable the Switch user button.	**static const DWORD EMBEDDED_DISABLE_BACK_BUTTON = 0x10**
Disable the Blocked Shutdown Resolver (BSDR) screen so that restarting or shutting down the system causes the OS to immediately force close any open applications that are blocking system shut down. No UI is displayed, and users are not given a chance to cancel the shutdown process	**static const DWORD EMBEDDED_DISABLE_BSDR= 0x20**

2. 기능 소개 & Demo

Custom Logon

- **Welcome Screen UI 숨기기 기능 제공**

XOR [edit]

A **bitwise XOR** is a binary operation that t
bits. The result in each position is 1 if only
the two bits are different, and 0 if they are

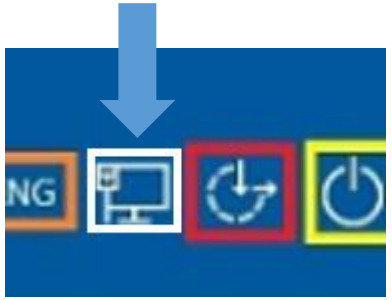
```
   0101 (decimal 5)
XOR 0011 (decimal 3)
  = 0110 (decimal 6)
```

Value	Description
1	Disables all Welcome screen UI elements.
2	Disables the **Power** button.
4	Disables the **Language** button.
8	Disables the **Ease of access** button.
16	Disables the **Switch user** button.
32	Disables the blocked shutdown resolver (BSDR) screen so that restarting or shutting down the system causes the OS to immediately force close any applications that are blocking system shut down. No UI is displayed and users are not given a chance to cancel the shutdown process. This can result in a loss of data if any open applications have unsaved data.

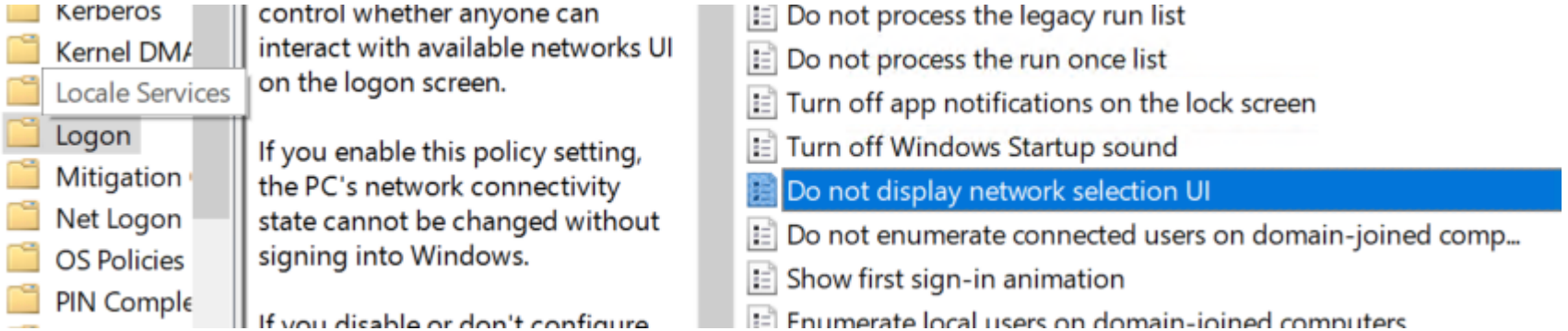
2. 기능 소개 & Demo

Custom Logon

- **Welcome Screen UI 숨기기 기능 제공**



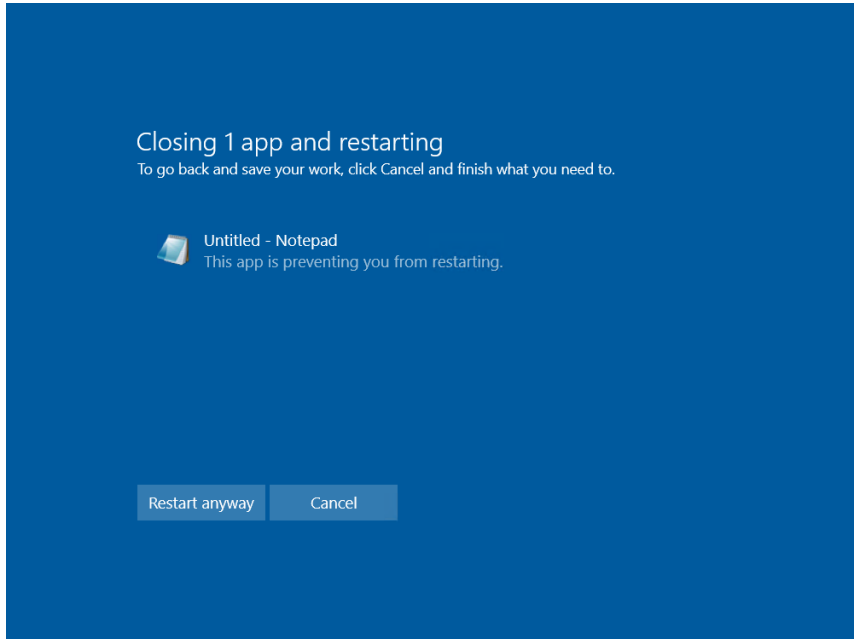
Local Group Policy (gpedit.msc)
Computer Configuration >> Administrative Templates >> System >> Logon
Do not display network selection UI 활성화



2. 기능 소개 & Demo

Custom Logon

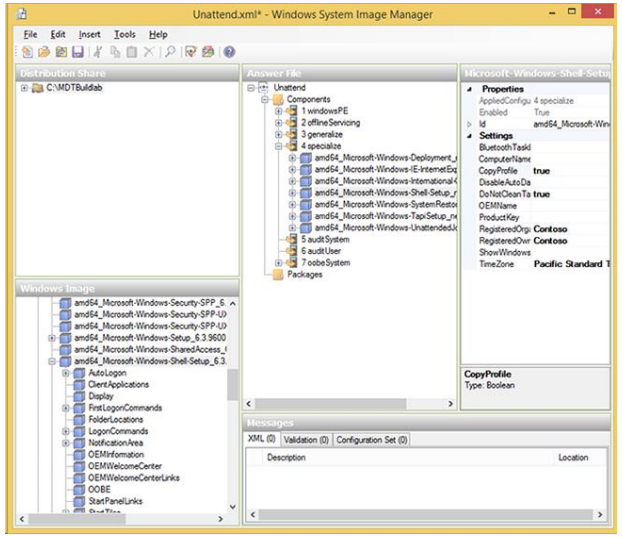
- 시스템 종료시 프로그램 종료 대기 화면 제거



2. 기능 소개 & Demo

Custom Logon

- **Windows SIM 이용한 Unattend에 Custom Logon 관련 설정 추가 가능**




```
<settings pass="specialize">
    <component name="Microsoft-Windows-Embedded-EmbeddedLogon" processorArchitecture="x86"
    publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <BrandingNeutral>17</BrandingNeutral>
      <AnimationDisabled>1</AnimationDisabled>
      <NoLockScreen>1</NoLockScreen>
      <UIVerbosityLevel>1</UIVerbosityLevel>
      <HideAutoLogonUI>1</HideAutoLogonUI>
    </component>
</settings>
```

Custom Logon

- **(TMI)**
  1511, Embedded Logon
  1607 and later, Custom Logon

Terminology

Turn on, enable: To make the setting available to the device and optionally apply the settings to the device. Generally *turn on* is used in the user interface or control panel, whereas *enable* is used for command line.

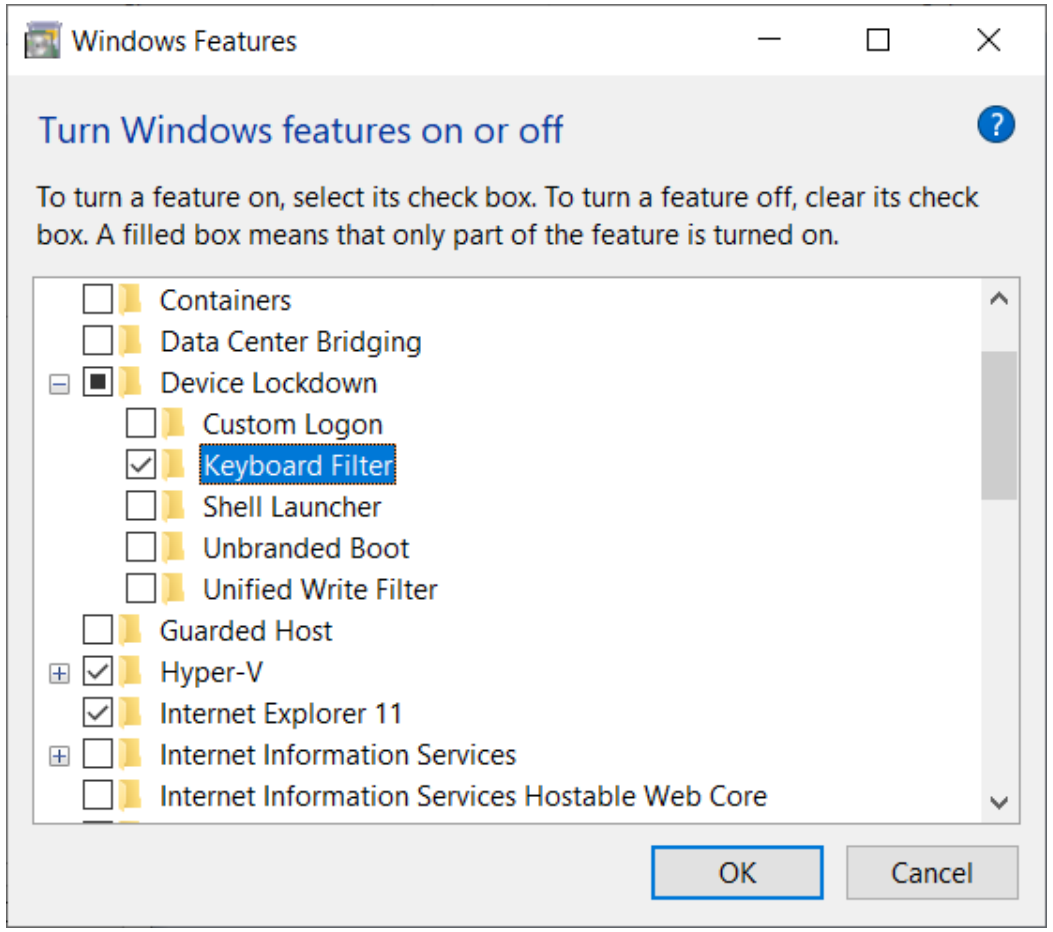
Configure: To customize the setting or sub-settings.

Embedded Logon: This feature is called Embedded Logon in Windows 10, version 1511.

Custom Logon: This feature is called Custom Logon in Windows 10, version 1607 and later.

2. 기능 소개 & Demo

Keyboard Filter



- **특정 키 또는 키 조합 동작 방지**
- **실제 키보드, 온스크린 키보드, 터치 키보드에서 필터링 가능**
- **멀티 키보드 조합으로 키 조합 입력 동작 방지**
  (ex. 실제 키보드에서 ctrl, alt 누른 상태에서 소프트웨어 키보드로 del키 또한 ctrl + alt + del로 인식)
- **(TMI)**
  1511, Embedded Keyboard
  1607 and later, Keyboard Filter
  WES7 Keyboard Filter

2. 기능 소개 & Demo

Keyboard Filter

- 특정 키 또는 키 조합 동작 방지

Keyboard scan codes and layouts

When a key is pressed on a physical keyboard, the keyboard sends a scan code to the keyboard driver. The driver then sends the scan code to the OS and the OS converts the scan code into a virtual key based on the current active layout. The layout defines the mapping of keys on the physical keyboard, and has many variants. A key on a keyboard always sends the same scan code when pressed, however this scan code can map to different virtual keys for different layouts. For example, in the English (United States) keyboard layout, the key to the right of the P key maps to "{". However, in the Swedish (Sweden) keyboard layout, the same key maps to "Å".

Keyboard Filter can block keys either by the scan code or the virtual key. Blocking keys by the scan code is useful for custom keyboards that have special scan codes that do not translate into any single virtual key. Blocking keys by the virtual key is generally more convenient because it is easier to read and Keyboard Filter suppresses the key correctly even when the location of the key changes because of a layout change.

When you configure Keyboard Filter to block keys by using the virtual key, you must use the English names for the virtual keys. For more information about the names of the virtual keys, see keyboard filter key names.

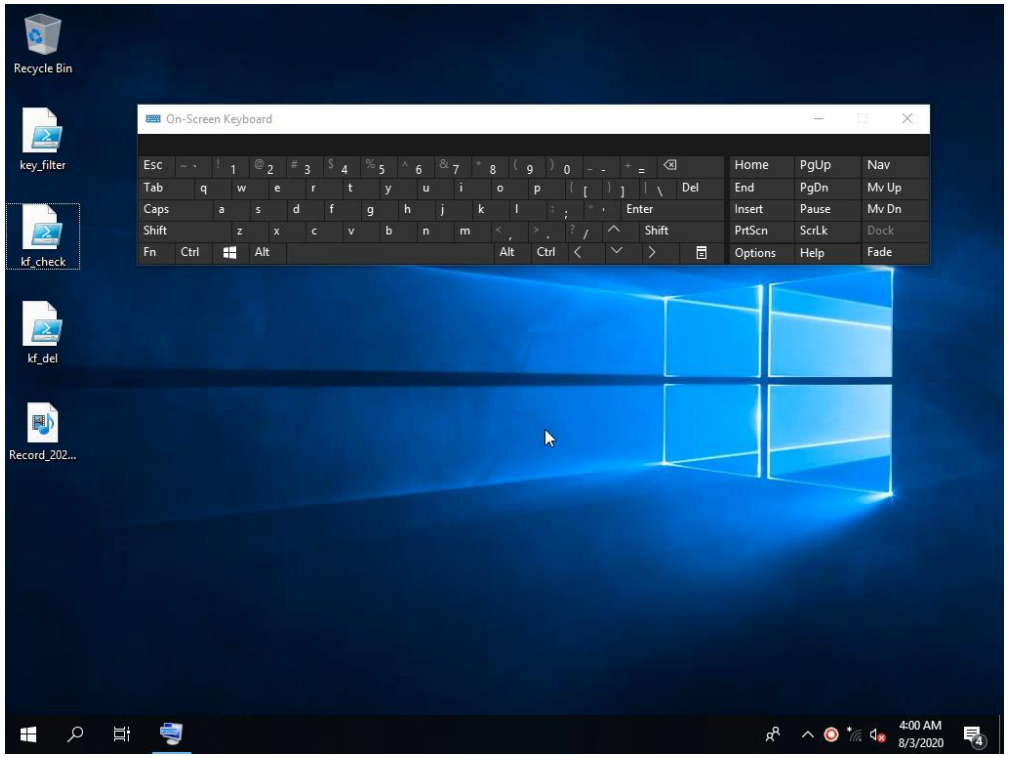
For the Windows on-screen keyboard, keyboard filter converts each keystroke into a scan code based on the layout, and back into a virtual key. This allows keyboard filter to suppress the on-screen keyboard keys in the same manner as physical keyboard keys, whether they are configured by scan code or virtual key.

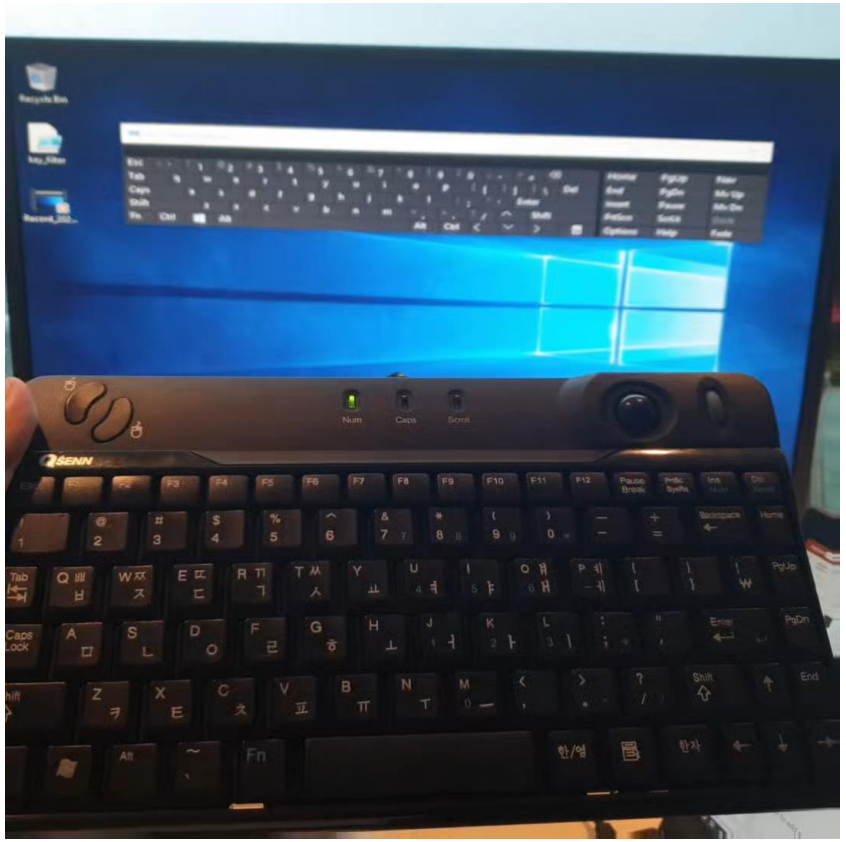
```
function Enable-Custom-Key($Id) {
    <#
    .Synopsis
        Toggle on a Custom Key keyboard filter Rule
    .Description
        Use Get-WMIObject to enumerate all WEKF_CustomKey instances,
        filter against key value "Id", and set that instance's "Enabled"
        property to 1/true.

        In the case that the Custom instance does not exist, add a new
        instance of WEKF_CustomKey using Set-WMIInstance.
    .Example
        Enable-Custom-Key "Ctrl+V"
        Enable filtering of the Ctrl + V sequence.
#>

    $custom = Get-WMIObject -class WEKF_CustomKey @CommonParams |
        where {
            $_.Id -eq "$Id"
        };

    if ($custom) {
# Rule exists.  Just enable it.
        $custom.Enabled = 1;
        $custom.Put() | Out-Null;
        "Enabled Custom Filter $Id.";

    } else {
        Set-WMIInstance `
            -class WEKF_CustomKey `
            -argument @{Id="$Id"} `
            @CommonParams | Out-Null
        "Added Custom Filter $Id.";
    }
}
```

2. 기능 소개 & Demo

Keyboard Filter

- 특정 키 또는 키 조합 동작 방지





Keyboard Filter

- **(TMI)**
  1511, Embedded Keyboard
  1607 and later, Keyboard Filter
  WES7 Keyboard Filter

Keyboard Filter Technical Reference (Standard 7 SP1)

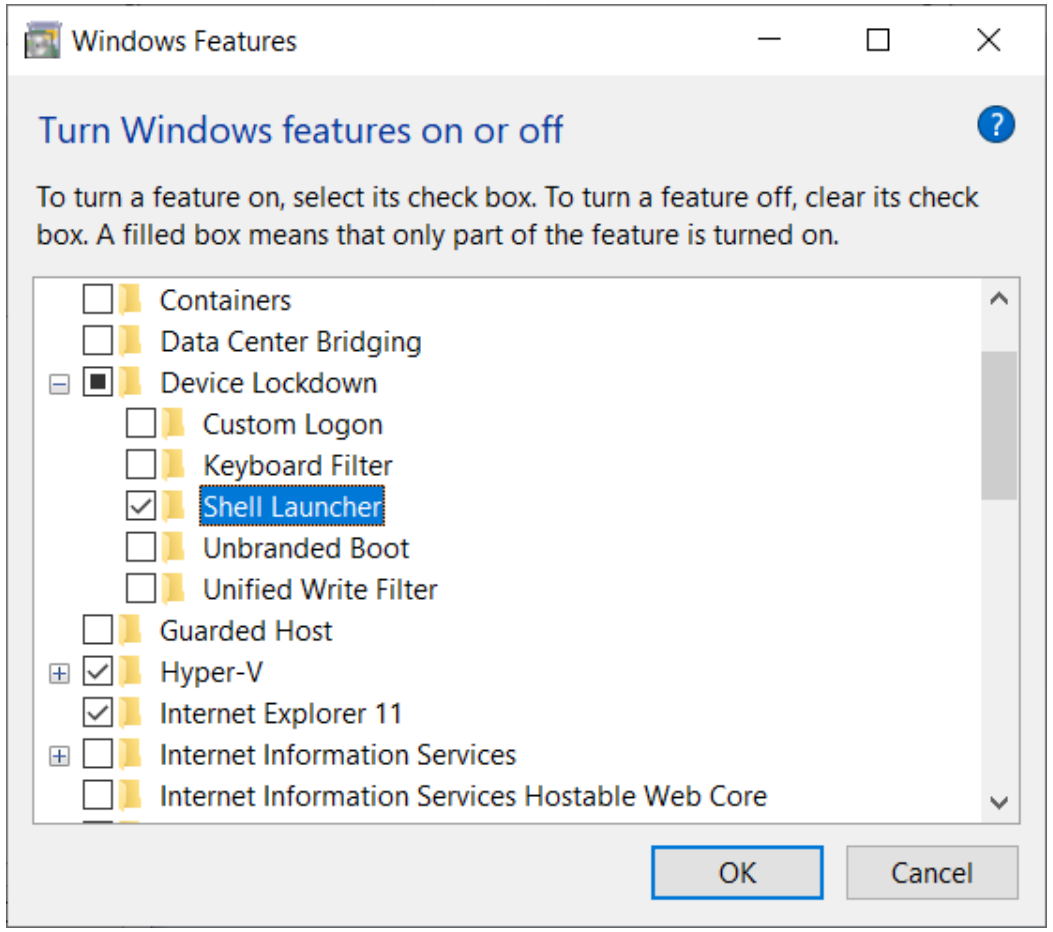
07/09/2014 • 2 minutes to read

7/8/2014

Keyboard Filter is an Embedded Enabling Feature (EEF) that enables you to block entry of unwanted keys or key combinations. Keyboard Filter includes many commonly used predefined key filters. In addition, you can easily create your own custom key filters for your embedded device.

2. 기능 소개 & Demo

Shell Launcher



- 기본 Windows 10 Shell을 custom shell로 변경 가능
- 각각의 유저 혹은 유저 그룹에 대하여 다른 application 적용 가능
- (TMI)
  1511, Embedded Shell Launcher
  1607 and later, Shell Launcher
  WES7 Custom Shell

2. 기능 소개 & Demo

Shell Launcher

- 기본 Windows 10 Shell을 custom shell로 변경 가능

PowerShell Copy

```powershell
# Check if shell launcher license is enabled
function Check-ShellLauncherLicenseEnabled
{
    [string]$source = @"
using System;
using System.Runtime.InteropServices;

static class CheckShellLauncherLicense
{
    const int S_OK = 0;

    public static bool IsShellLauncherLicenseEnabled()
    {
        int enabled = 0;

        if (NativeMethods.SLGetWindowsInformationDWORD("EmbeddedFeature-ShellLauncher-Enabled", out er
            enabled = 0;
        }

        return (enabled != 0);
    }

    static class NativeMethods
    {
        [DllImport("Slc.dll")]
        internal static extern int SLGetWindowsInformationDWORD([MarshalAs(UnmanagedType.LPWStr)]strir
    }

}
"@
```

2. 기능 소개 & Demo

Shell Launcher

- 각각의 유저 혹은 유저 그룹에 대하여 다른 application 적용 가능

2. 기능 소개 & Demo

Shell Launcher

- **(TMI)**
  1511, Embedded Shell Launcher
  1607 and later, Shell Launcher
  WES7 Custom Shell

Terminology

- **Turn on, enable:** To make the setting available to the device and optionally apply the settings to the device.
- **Configure:** To customize the setting or sub-settings.
- **Embedded Shell Launcher:** This feature is called Embedded Shell Launcher in Windows 10, version 1511.
- **Custom Shell Launcher:** This feature is called Shell Launcher in Windows 10, version 1607 and later.

Custom Shell (Standard 7 SP1)

07/09/2014 • 2 minutes to read

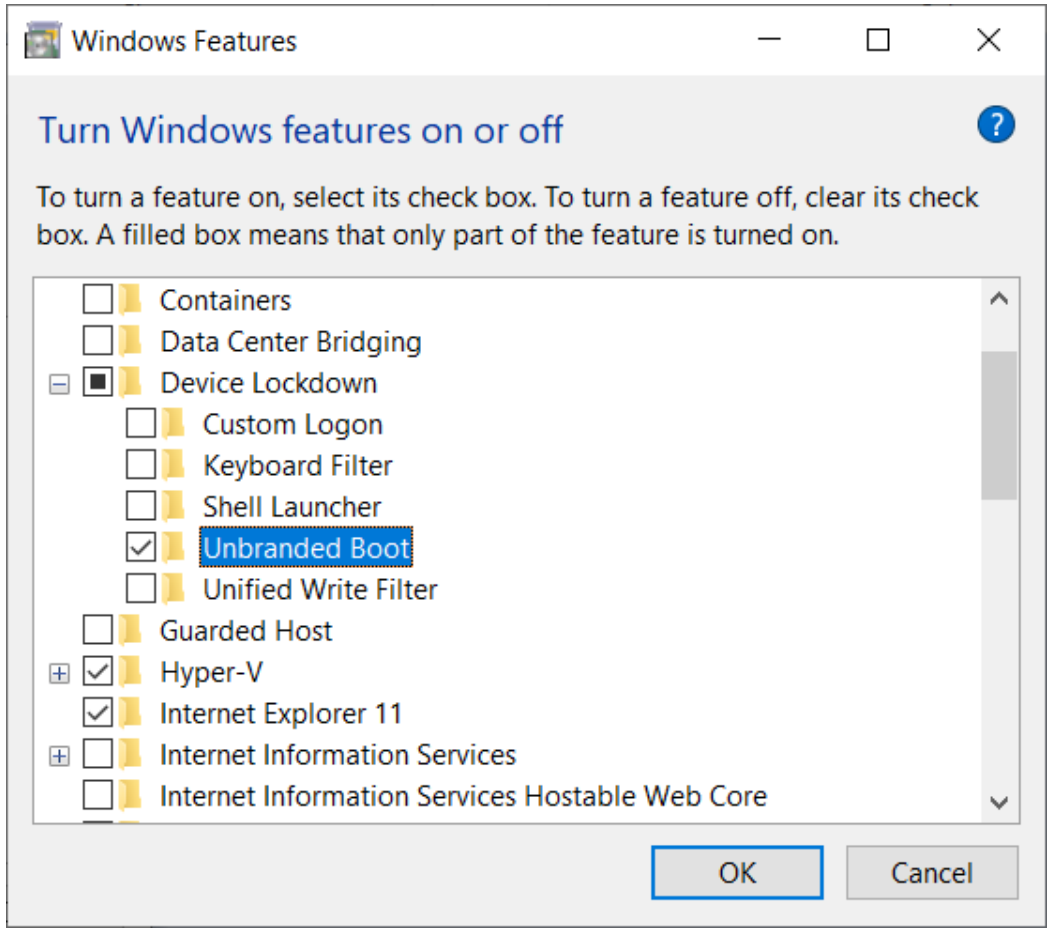
7/8/2014

This section describes how to create a custom shell experience by using the custom shell components. Each component can be used separately or in any combination. By using a custom shell, you can do the following:

- Remove boot screens
- Use Shell Launcher to reference a custom shell
- Remove Windows branding from startup screens
- Replace the background image for startup screens
- Add message blockers

2. 기능 소개 & Demo

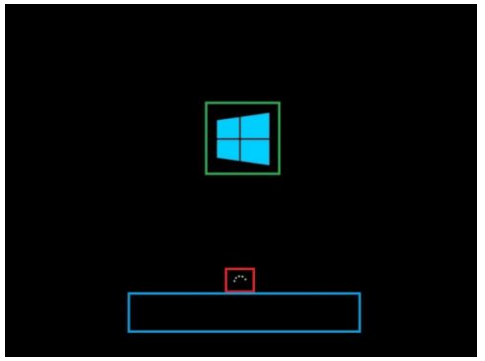
Unbranded Boot



- **Windows 시작 혹은 resume시 나타나는 elements 제거**
- **Windows 오류 발생시 나오는 crash screen 제거**
- **(TMI)**
  1511, Embedded Boot Experience
  1607 and later, Custom Boot Experience

2. 기능 소개 & Demo

Unbranded Boot



- **Windows 시작 혹은 resume시 나타나는 elements 제거**

HideAllBootUI	Contains an integer that suppresses all Windows UI elements (logo, status indicator, and status message) during startup. Set to 1 to suppress all Windows UI elements during startup; otherwise; set to 0 (zero). The default value is 0.
HideBootLogo	Contains an integer that suppresses the default Windows logo that displays during the OS loading phase. Set to 1 to suppress the default Windows logo; otherwise; set to 0 (zero). The default value is 0.
HideBootStatusIndicator	Contains an integer that suppresses the status indicator that displays during the OS loading phase. Set to 1 to suppress the status indicator; otherwise; set to 0 (zero). The default value is 0.
HideBootStatusMessage	Contains an integer that suppresses the startup status text that displays during the OS loading phase. Set to 1 to suppress the startup status text; otherwise; set to 0 (zero). The default value is 0.

Unbranded Boot

- **Windows 오류 발생시 나오는 crash screen 제거**

Configure Unbranded Boot using Unattend

You can also configure the Unattend settings in the Microsoft-Windows-Embedded-BootExp component to add Unbranded Boot features to your image during the design or imaging phase. You can manually create an Unattend answer file or use Windows System Image Manager (Windows SIM) to add the appropriate settings to your answer file. For more information about the Unbranded Boot settings and XML examples, see the settings in Microsoft-Windows-Embedded-BootExp.

Unbranded Boot settings

The following table shows Unbranded Boot settings and their values.

Setting	Description
DisableBootMenu	Contains an integer that disables the F8 and F10 keys during startup to prevent access to the Advanced startup options menu. Set to 1 to disable the menu; otherwise; set to 0 (zero). The default value is 0.
DisplayDisabled	Contains an integer that configures the device to display a blank screen when Windows encounters an error that it cannot recover from. Set to 1 to display a blank screen on error; otherwise; set to 0 (zero). The default value is 0.

Unbranded Boot

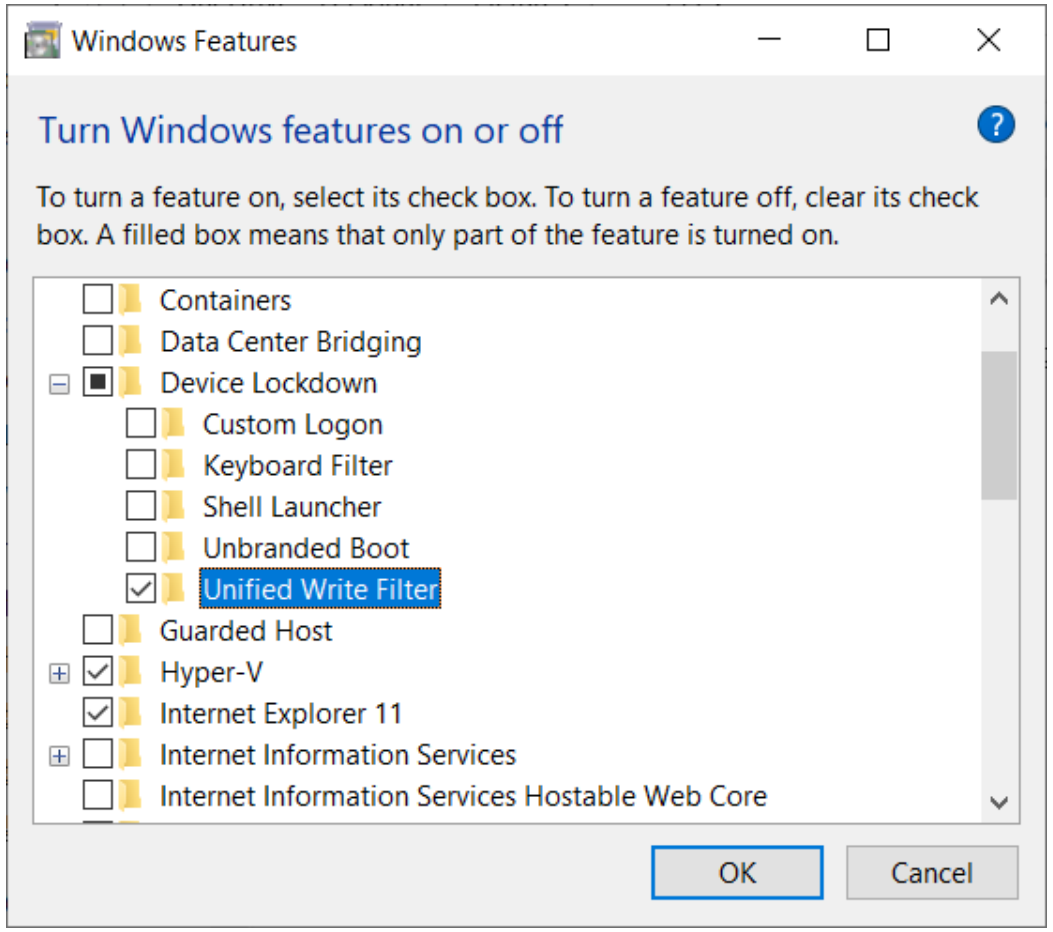
- **(TMI)**
  1511, Embedded Boot Experience
  1607 and later, Custom Boot Experience

Terminology

- **Turn on, Enable:** To make the setting available to the device and optionally apply the settings to the device. Generally "turn on" is used in the user interface or control panel, whereas "enable" is used for command line.
- **Configure:** To customize the setting or sub-settings.
- **Embedded Boot Experience:** this feature is called "Embedded Boot Experience" in Windows 10, build 1511.
- **Custom Boot Experience:** this feature is called "Custom Boot Experience" in Windows 10, build 1607 and later.

2. 기능 소개 & Demo

Unified Write Filter



- 가상 오버레이를 이용한 쓰기 방지 필터
- Hdd, ssd, internal usb devices, external sata 보호 가능
- Disk, ram overlay 방식 지원
- (TMI)
  WES7 ewf + fbwf

2. 기능 소개 & Demo

Unified Write Filter

- 가상 오버레이를 이용한 쓰기 방지 필터

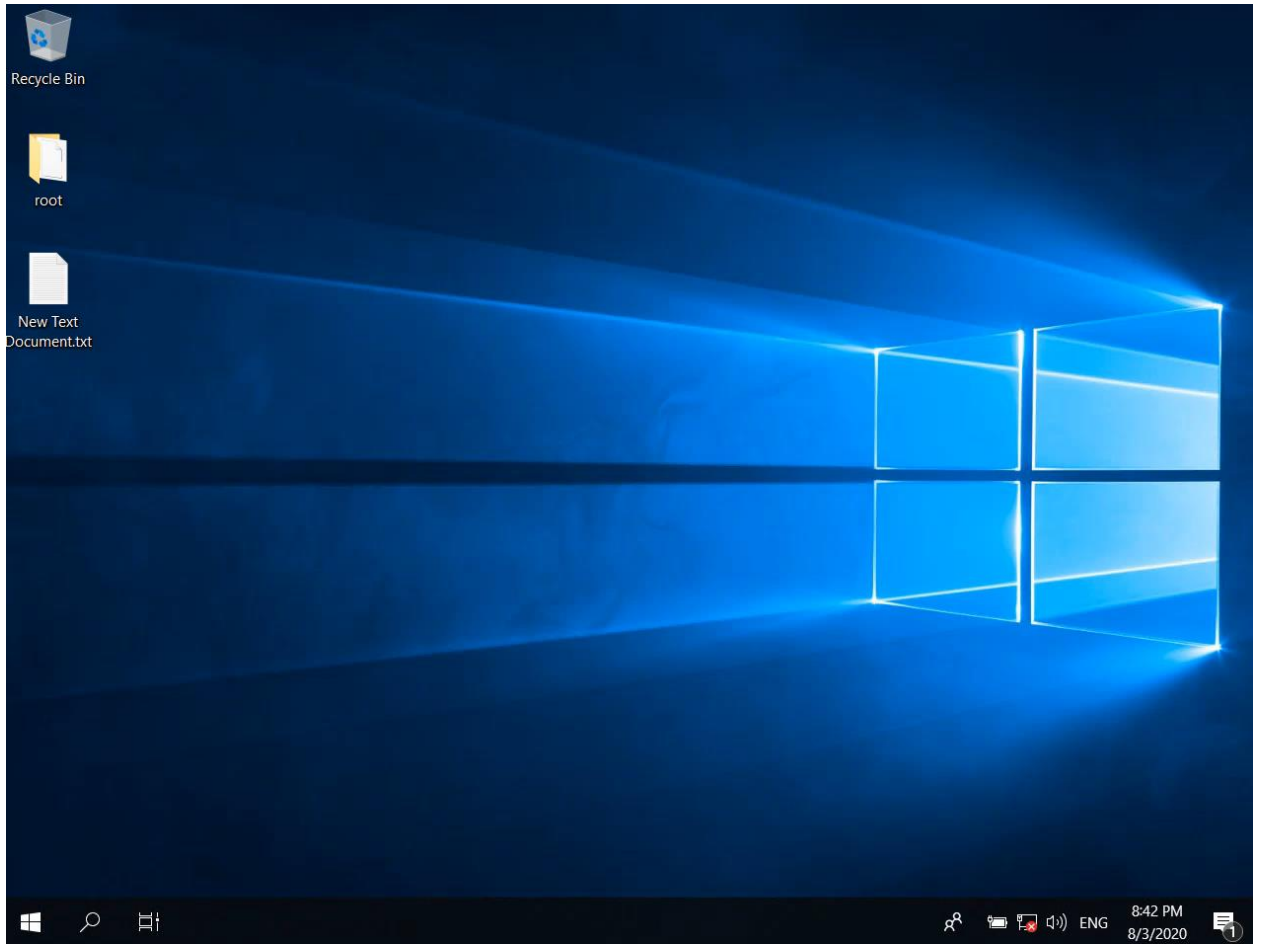
```
Current Session Settings


FILTER SETTINGS
    Filter state:    OFF
    Pending commit:  N/A
    Shutdown pending:No

SERVICING SETTINGS
    Servicing State: OFF

OVERLAY SETTINGS
    Type:                RAM
    Maximum size:        1024 MB
    Warning Threshold:   512 MB
    Critical Threshold:  1024 MB
    Freespace Passthrough: OFF
    Persistent: OFF
    Reset Mode: N/A



VOLUME SETTINGS
    *** No volumes configured


REGISTRY EXCLUSIONS
    *** No exclusions
```

```
Next Session Settings


FILTER SETTINGS
    Filter state:    ON
    Pending commit:  N/A

SERVICING SETTINGS
    Servicing State: OFF

OVERLAY SETTINGS
    Type:                Disk
    Maximum size:        1024 MB
    Warning Threshold:   512 MB
    Critical Threshold:  1024 MB
    Freespace Passthrough: ON
    Persistent: OFF
    Reset Mode: N/A



VOLUME SETTINGS
Volume fc96e861-aa4d-4b66-bd66-09055b4f72dd [C:]
    Volume state:     Protected
    Volume ID:        fc96e861-aa4d-4b66-bd66-09055b4f72dd

    File Exclusions:
Next Session Exclusions for Volume fc96e861-aa4d-4b66-bd66-09055b4f72dd [C:]
    C:\Users\HancomMDS\Desktop\root\Unprotected



REGISTRY EXCLUSIONS
    *** No exclusions
```

2. 기능 소개 & Demo

Unified Write Filter

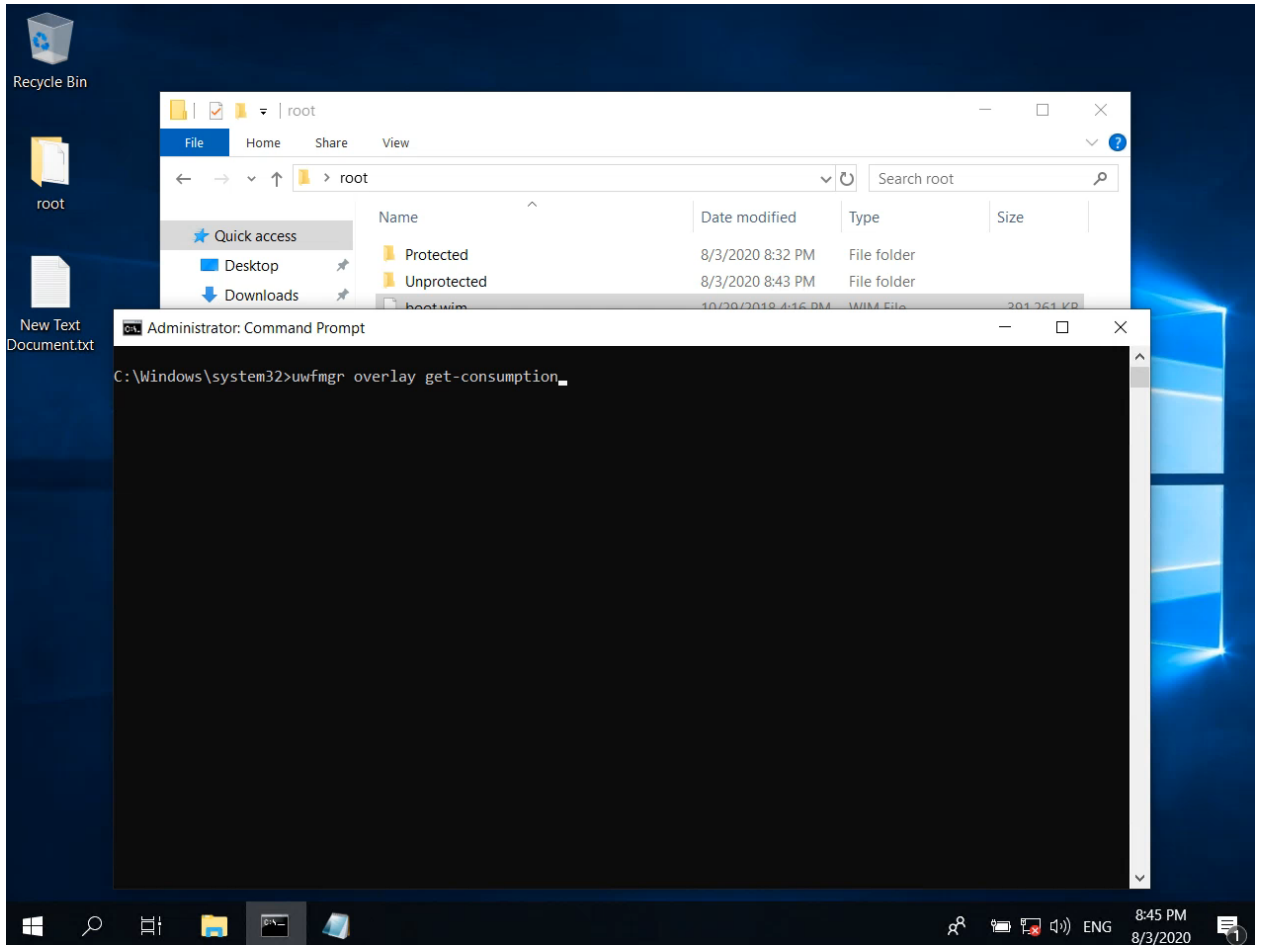
- 가상 오버레이를 이용한 쓰기 방지 필터



2. 기능 소개 & Demo

Unified Write Filter

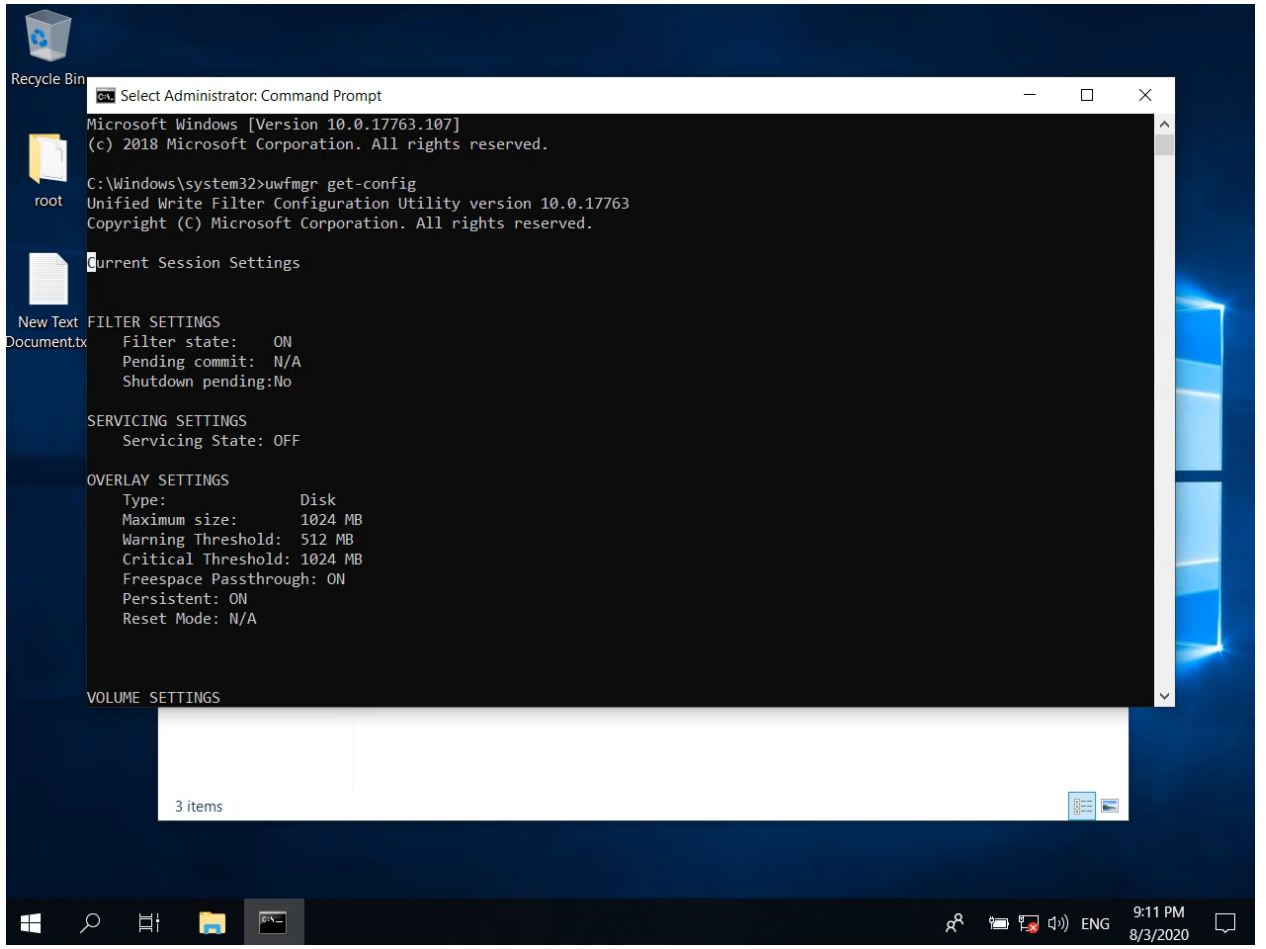
- 가상 오버레이를 이용한 쓰기 방지 필터



2. 기능 소개 & Demo

Unified Write Filter

- 가상 오버레이를 이용한 쓰기 방지 필터



Unified Write Filter

- **Hdd, ssd, internal usb devices, external sata 보호 가능**

- UWF can protect most supported writable storage types, including physical hard disks, solid-state drives, internal USB devices, and external SATA devices. You cannot use UWF to protect external removable drives, USB devices or flash drives. Supports both master boot record (MBR) and GUID partition table (GPT) volumes.

Unified Write Filter

- **Disk, ram overlay 방식 지원**

UWF overlay

You can choose where the overlay is stored (RAM or disk), how much space is reserved, whether the overlay persists after a reboot.

To increase uptime, set up monitoring to check if your overlay is filling up. At certain levels, your device can warn users and/or reboot the device.

To learn more, see UWF Overlay location and size.

Unified Write Filter

- **(TMI)**
  WES7 ewf + fbwf

Enhanced Write Filter with HORM (Standard 7 SP1)

07/09/2014 • 2 minutes to read

7/8/2014

Enhanced Write Filter (EWF) lets you write-protect a run-time image. By redirecting all write requests to RAM, EWF enables the run-time image to maintain the appearance of a writable run-time image.

File-Based Write Filter (FBWF) (Standard 7 SP1)

07/09/2014 • 2 minutes to read

7/8/2014

File-Based Write Filter (FBWF) enables redirection of all changes made to a protected volume to an in-memory overlay. This provides system designers with the option of discarding all changes made to a system in a manner that is transparent to user applications. This inherent feature of FBWF can also be used for booting Windows Embedded Standard 7 images from read-only flash media.

Q & A

Thank you