Windows "Desktop"

- Windows 10 Usability Tweaks
- STUC Valley
- Installing Windows 11 (from ISO) on real hardware (without SB and/or TPM 2.0 support)

Windows 10 Usability Tweaks

A subjective guide based on supposedly-objective data

This guide aims to collect a number of obvious and less-obvious tweaks on making Windows 10 a more usable interactive system. Any entry should be backed by direct, measurable data and external references: no placebo!

Shell performance

win32kbase!*Crit

A common cause of bad interactive performance (and gaming microstutters!) due to Win32 message loop lock contention. Allegedly regressed around RS3. ETW tracing with context switches should provide enough information regarding this.

References: Bruce Dawson's blog series

Explorer side-panel view

Common contention originating from callbacks in Windows.Storage.dll due to change notifications if any drives (e.g. hot-pluggable SATA drives or external drives) are expanded in the side view and any change is made to a directory shown there. Worsens with many File Explorer windows.

Tweaks

1. Disable shell side panel in toolbar. Might take some closing/reopening windows to get the preference to save.
2. Ideally: hide removable drives from the side view so they do not auto-expand. Anything under 'This PC' already does not, only removable drives somehow have the tendency to do so. **TODO: someone find this**

Task Manager 'Details' tab

Icon refreshing seems to be extremely slow. Try to avoid refreshing the Details tab when running Task Manager in the background.

Global performance

Memory compression

Memory compression is regressive on high-memory systems (especially with SSD page file backing) as decompressing page-in still leads to unneeded *blocking* latency (as opposed to disk ops from failed disk cache generally being asynchronous). In addition to that, it confuses system-wide memory metrics as the commit usage for compressed memory is duplicated: once in the process that's compressed, then yet again as `CM31` paged pool.

References: microsoft/WinDev#35

Tweaks

1. `Disable-MMAgent -mc` in PowerShell.
2. Reboot.

Svchost split

Improves reliability, but has been reported to also affect interactive performance. No references at this point.

STUC Valley

Stupid, **T**echnically **U**seless, **C**onfusing

*Those are a few words that come to mind once you see **what in the fresh hell** happened to Velocity feature names once Sun Valley development picked up the pace*

Here's a table of acronyms used in various feature names

Acronym	Explanation
AL(L)B	Animated Lock/Logon Background
AUXEMP	Add Undocked Xaml Extensions Metadata Provider
CBR(S)	Cloud Backup Restore Settings
CUIDFS	Credential UI Desktop Fluent Styling
CUIPN	Credential UI Provider Name
DGI	Dashboard Gesture Integration
DI	Drag Improvements
DLOOBE	Desktop Lite Out Of Box Experience
DTC	Deprecate Tile Control
Emrbspi	Enable Monitor Removal Behavior Spi
Erpsrbspi	Enable Restore Previous State Recalc Behavior Spi
FTVTM	Frontload Timeline Visual Tree Modifications
MTCMCTSA	Move Taskbar Context Menu Options(?) To Settings App
MW	Minimize Windows
RW	Restore Windows
SA	Snap Assist
S(BT)G	Snap Based Task Groups
SE	Snap Education
SSWPU	Shell Set Window Position Unrestricted
Swonms	Shrink Window On Move Size
TD	Taskbar Dll
TTP	Taskbar Test Proxy
TUC	Taskbar Undocked Controls

Acronym	Explanation
UDTHD	Use Desktop Taskbar Handler Dll
UTTV	Updated Taskview Timeline Visuals
UUS	Undocked Update Stack
V(D)PM	Virtual Desktops Per Monitor
VTRRCC	Visual Tree Rounded Rectangle Cpu Clip
WTA	Win32 Taskbar Access
WUIOD	WinUI On Desktop

Acronym	Guess
HFEFNM	Hello First Enrollment Flags N M
GTG	G Touch Gesture
VDS	Virtual Desktop S
VDR	Virtual Desktop Rename
VDW	Virtual Desktop W
MSZ	?
CUICOORD	Credential UI C O O Rounded Dialog
OP	
XHSWUI	
SCM	
STUC	S Taskbar Undocked Controls
NP	
NM	
VT	Visual Tree
SR	
DLNP	Desktop Lite N Privacy
Tr	
AT	
P	Palette
IA	
V	
WS	
CAC	

Acronym	Guess
OSetup	
63	
DA	
LC	
SB	
LA	
SN	
DE	
HVCZM	
DLOMETRIC	Desktop Lite O M E T R I C
HVCHM	

Installing Windows 11 (from ISO) on real hardware (without SB and/or TPM 2.0 support)

Created: June 20, 2021	Last Update: June 20, 2021
Authors:	Gustave Monce, Daniel Kornev
Status:	**Draft**
Supported Windows 11 Version:	21996.1

WIP

Overview

Here's some quick notes on installing Windows 11 on real hardware from ISO in case when hardware doesn't support TPM and/or SB (Secure Boot). For this example, we're assuming a system with *no* other critical disks installed, and a helpful host system being around to set up the initial image.

This guide has been created for the 21996.1 version of Windows 11.

Prerequisites

Host

- Windows 10 Iron or Cobalt (20279 or 21xxx+) - though 20H2 can also work just fine
- Utility USB flash drive of ~8GB+

Target

- Boot drive larger than 50 GiB

Common: Files & Tools

1. Download Windows 11 ISO
2. Download Windows USB/DVD Tool
3. Install it

Prepare USB Drive for Flashing

1. Get an empty USB Flash drive (with size at least 8GB)
2. Use Windows USB/DVD Tool
3. Follow its instructions to flash your Windows 11 ISO onto the USB Flash drive

Apply Your Image

Clean Disk & Lay Out Partitions

1. Boot from USB Flash drive
2. Open **diskpart**
3. Type this:

```
list disk
```

Take a note of the name of your device's main disk drive, e.g., `disk 0` 4. Type this (where `disk 0` is your device's main disk drive):

```
select disk 0
clean
```

5. Follow the instructions for UEFI-based PCs from docs.microsoft.com to properly lay out your device's main disk for Windows 11 installation:

```
rem == CreatePartitions-UEFI.txt ==
rem == These commands are used with DiskPart to
rem    create four partitions
rem    for a UEFI/GPT-based PC.
rem    Adjust the partition sizes to fill the drive
rem    as necessary. ==
select disk 0
clean
convert gpt
rem == 1. System partition =========================
create partition efi size=100
rem    ** NOTE: For Advanced Format 4Kn drives,
rem              change this value to size = 260 **
format quick fs=fat32 label="System"
assign letter="S"
rem == 2. Microsoft Reserved (MSR) partition =======
create partition msr size=16
rem == 3. Windows partition ========================
rem ==    a. Create the Windows partition ==========
create partition primary
rem ==    b. Create space for the recovery tools ===
rem       ** Update this size to match the size of
rem          the recovery tools (winre.wim)
rem          plus some free space.
shrink minimum=500
rem ==    c. Prepare the Windows partition =========
format quick fs=ntfs label="Windows"
assign letter="W"
rem === 4. Recovery partition ======================
create partition primary
format quick fs=ntfs label="Recovery"
assign letter="R"
set id="de94bba4-06d1-4d40-a16a-bfd50179d6ac"
gpt attributes=0x8000000000000001
list volume
```

6. Mark your Windows partition as active in diskpart:

```
list partition
note the name of your Windows partition (e.g., "partition 3")
```

```
select partition 3
active
```

7. Type **exit** to leave **diskpart**

Apply Image

1. Check which OS SKU you want to install:

```
dism /Get-WimInfo /WimFile:D:\Sources\install.wim <— here D:\ is the drive name of your USB drive
```

2. Remember its index and use it to apply its image to your Windows partition (e.g., "W"):

```
dism /Apply-Image /ImageFile:D:\Sources\install.wim /index:1 /ApplyDir:W:\
```

3. Create boot records:

```
W:\Windows\System32\bcdboot G:\Windows
```

If you'll see "Failure when attempting to copy boot files" error message then use this command instead:

```
W:\Windows\System32\bcdboot c:\windows /s s: /f ALL
```

4. Type **exit** in the **Command Prompt window**

Load Windows 11

At this time, your Windows 11 will boot. If everything is correct you should see the OOBE.