דסב

SSD Script Patent

This script is unique in that it runs totally headless (No user intervention)
The script starts by

1. identifying system is either a MBR or UEFI
2. Based on results in A. it runs a unique script to manage each of the system types
3. It then moves on to identifying the SSD and formatting it (either GPT for UEFI or MBR for BIOS. Results gathered in A.) The script does a partition byte comparison before formatting the drive to ensure it never ever formats any other drive then the SSD which it is running off.
4. After the format is complete the script find the Windows partition on the HDD and copies THE FILES over to the newly created windows partition on the SSD (this is a crucial element in which all other cloning software differ. Some cloning software either do what is called a bare metal clone in which the source drive is literally written to the destination drive right down to the binary level. The problem with this approach is A. the destination drive will have to either be equal or greater to source disk. And B. If the destination disk is larger the partition size will still mirror that of the old one (for example if I clone a 256GB drive on to a 512GB drive, I will only be able to use 256GB of the 512GB drive since the partition size of the original drive was only 256GB ). Another problem is that since it has to copy over every byte of data from the source drive it will always be transferring the maximum amount of space from the source disk IRRELEVANT to how much of the space is actually used. This not only increases time it also puts extra strain on both disk wearing them out ultimately causing them to fail. This is especially true for SSD which do not have as long a Read/Write tolerance as HDD. Another method is Imaging which is the process of saving the entire contents of a partition and placing it in to an IMAGE file. Now imaging is great in a corporate environment in which you may need multiple copies of the same exact system. However in the personal computing sector this is ultimately pointless and is A. an extra step B. requires a whole additional drive to store the image on and C. is prone to image corruption and data loss.
5. The last step is installing the bootloader which again differs depending on the system type (UEFI or MBR)
6. The script then automatically reboots and starts up the newly cloned drive.

For all the reasons stated above I’m sure you’ll understand that our script the fastest, safest and cleanest method of cloning a windows drive. What really sets us apart is the fact the there is no potential even for user intervention or mistakes. If the script does hit a stone wall it instantly reboots the machine.