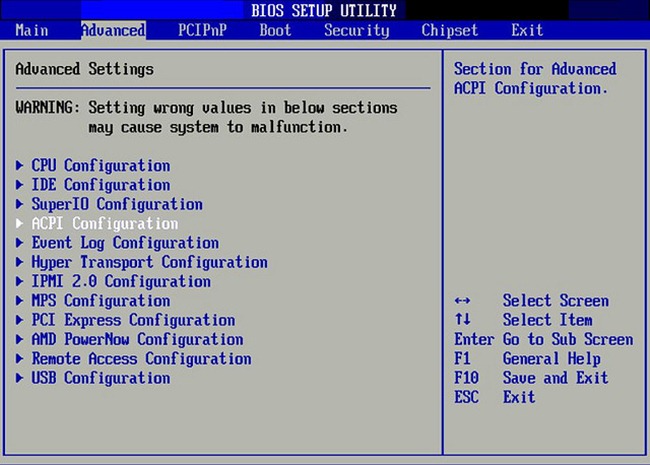
**[HTG Explains: Learn How UEFI Will Replace Your PC’s BIOS](http://www.howtogeek.com/56958/htg-explains-how-uefi-will-replace-the-bios/" \o "HTG Explains: Learn How UEFI Will Replace Your PC’s BIOS)**



While most people may be familiar with a PC’s BIOS, they may not know what it is or what it does. Since UEFI is poised to take its place, let’s take a geek retrospective on these two  technologies.

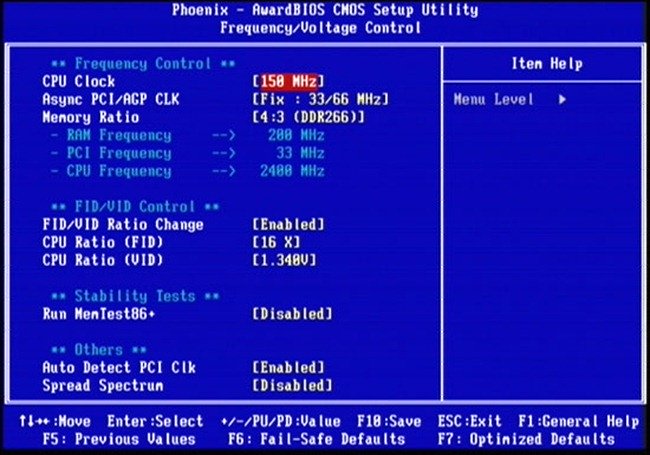
Understanding BIOS (Basic Input/Output System)



(Image from [richardmasoner](http://www.flickr.com/photos/bike/189646022/sizes/z/in/photostream/))

The BIOS (pronounced “bye-ose”) is a computer’s Basic Input-Output System. It’s a low-level software that’s so important integrally that resides on a chip that’s built into the motherboard. When your computer starts up, it’s the BIOS’s job to wake up the various components and make sure they’re functioning, then it passes off functionality to your operating system or another boot loader. A “long” time ago, computers used very different and proprietary ways to do this, but over time IBM’s 5150 sort of became a standard on which to base hardware compatibility. It used the Intel 8088 processor which was 16-bit, and so the BIOS itself is 16-bit and is allowed 1MB of address space. It also uses a Master Boot Record, or MBR, to specify the computer’s partition table, which in turn tells the BIOS where the operating system is.

POSTing is a BIOS-governed process. It’s a power-on self-test which checks the validity and correct functionality of your components. If something is wrong, you’ll see an error displayed or hear a cryptic series of beep codes. You may also see the acronym CMOS. This refers to the battery-backed memory which is used in conjunction with the non-volatile RAM used with the BIOS. It’s actually not accurate anymore, since this method has been replaced with flash memory (also referred to as EEPROM) in contemporary systems.



All in all it was a good system. It provided an interface from which you could enable/disable individual components and advanced hardware options. Of course, as with everything in the silicon world, BIOS obsolesced in comparison to the newer hardware that was developing. In order to make up for a lot of its shortcomings, extensions were developed. The prime example of this is Advanced Configuration and Power Interface, or ACPI, which was put in place to help device configuration and advanced power management functions. But, over time, it’s limitations needed to be overcome by a brand new system.

The Successor: UEFI

The Unified Extensible Firmware Interface, also known as UEFI (pronounced “oofy” or “U effy”) was originally developed by Intel to circumvent these issues for its 64-bit Itanium-based servers. Over time, it was renamed from the Intel Boot Initiative and given over to the Unified EFI Forum, which governs the specifications for it. UEFI brought some major changes to the pre-boot environment.

**Visuals**

Many implementations of EFI just look like the traditional BIOS, but others customize the visual layout fundamentally. While pretty pictures aren’t really necessary, they can be really helpful for things like overclocking. Take a look at the differences:



As you can see, it’s easy to use, mouse-capable, and has a lot of potential.



MSI calls its version the “EFI Click BIOS.”

**16-bit vs 32- and 64-bit**

While the BIOS is limited to 16-bit processes and 1MB of memory addressing, UEFI isn’t constrained in that way. It can function in 32-bit and 64-bit modes, allowing much more RAM to be addressed by more complex processes. It also can be architecture independent and provide drivers for components that are also independent of what kind of CPU you have.

**Booting**

The MBR is limited to 4 primary partitions per disk and bootable disks are limited in size to 2.2 TB. UEFI uses the GUID Partition Table, which utilizes Globally Unique IDs to address partitions and allows booting from hard disks as large as 9.4 ZB. A terabyte (technically, a tebibyte) is 1024 GB, and a zettabyte (zebibyte) is 1024x1024x1024 GB. Seems pretty future-proof for the moment, no? And the benefits don’t stop there; UEFI allows more boot options, doesn’t prescribe particular file systems, and has excellent network booting abilities. OS boot loaders can also serve as extensions to the UEFI, which itself can function as a proper boot loader.

**Extensions**

UEFI supports older extensions, like ACPI, which aren’t  dependent on a 16-bit runtime environment. Cryptic beep-code errors are also a thing of the past, as extensions can better test components (unless something more dire is wrong, like a bad processor). In addition, it support EFI-based partitions on hard disks which manufacturers can use to add more functionality. Asus’s Splashtop instant-on OS is a good example. While it works with BIOS, UEFI can offer better boot times and loading for something like that.

From BIOS to UEFI



Its potential is what makes UEFI so promising. Not everything is optimal yet, but it operates already at BIOS level, so manufacturers have been starting to use it more on their motherboards. They can use older extensions with the new system until newer ones can take over, and the older visual style can also be used in the interim. The switch has been rolling out at an ever-increasing pace over the past few years. It’s not possible to put UEFI on BIOS-based motherboards, but odds are that when you buy a new system or perform an upgrade in the next few years you’ll be making the switch, perhaps without even realizing it. It’s a slow switch, but it seems to be an inevitable one.