**Phase 3: Driver Execution Environment (DXE):**

Driver Execution Environment (DXE) is the phase after PEI in UEFI boot flow. The DXE phase is the phase where most of the system initialization is performed. The PEI phase is responsible for initializing permanent memory so that the DXE phase can be loaded and executed. The state of system at the end of PEI phase is passed to DXE in the form of HOBs.

The DXE phase consist of following components.

1. DXE Core:

The DXE core produces a set of Boot Services, Runtime Services, and DXE services.

1. DXE Dispatcher:

The DXE dispatcher is responsible for discovering and executing DXE drivers in correct order.

1. DXE Drivers

These components work together to initialize the platform and provide services required to boot to an OS. The DXE phase is terminated when an OS successfully begins its boot process that is, when the BDS phase starts. Only the runtime services provided by DXE can run into OS.

DXE Core:

DXE core is designed to be completely portable with no processor, chipset or platform dependencies.

The DXE phase initially depends on HOB list from its previous phase PEI. This means DXE phase doesn’t depend on any services from a previous phase, so all the prior phases can be unloaded once the HOB list is passed to the DXE core.

The DXE core doesn’t contain any hard-coded addresses. As a result, the DXE phase can be loaded anywhere in physical memory.

The DXE core produces EFI System Table and its associated EFI Boot Services and EFI Runtime Services. EFI Boot Services can be terminated once OS starts. EFI System Table and EFI Runtime Services can continue in the OS runtime environment

The DXE core also contains DXE dispatcher, who main purpose is to discover and dipath DXE Drivers.

1. Hand of Blocks (HOBs):
2. The HOB list contains all the information that the DXE core requires to produce it memory-based services. The HOB list contains boot mode, processor’s instruction set, memory discovered in the PEI phase, and the firmware volumes. The firmware volumes may contain DXE drivers.
3. The first entry in the HOB list is always a PHIT block. PHIT is for Phase Handoff Information Table, it contains boot mode.
4. DXE architectural Protocols:

The DXE core is abstracted from the platform hardware through a set of DXE architectural protocols. These protocols provide EFI Boot Services and EFI Runtime Services.

1. EFI System Table:

The DXE core produces EFI System Table, which contains all the information that is required for every DXE drivers and executable images invoked by BDS to use these services provided by DXE core.

1. EFI Boot Services Table:
2. Task Priority Services
3. Memory Services
4. Event and Timer Services
5. Protocol Handler Services
6. Image Services
7. Driver Support Services
8. EFI Runtime Services Table:
9. Variable Services
10. Real Time Clock Services
11. Reset Services
12. Status Code Services
13. Virtual Memory Services
14. DXE Service Table
15. Global Coherency Domain Services
16. DXE Dispatcher Services

DXE Dispatcher:

After the DXE Core is initialized, control is handed to the DXE dispatcher. The DXE dispatcher is responsible for discovering and dispatching DXE drivers found in firmware volumes described by HOB list. As execution continues, DXE dispatcher searches for DXE drivers in other firmware volumes might be located.

Each firmware volumes contain a priori file, used to indicate the order in which DXE drivers are loaded and executed. A firmware file may not have a priori file, then dependency expression is used to load and execute DXE drivers. A priori file provides strongly ordered list of DXE drivers that do not need to use dependency expression. A firmware volume may not provide a priori file, in that case dependency expression is evaluated in order to load execute DXE drivers.

DXE Drivers:

DXE drivers are of two types.

1. DXE drivers that execute early in DXE phase
2. DXE drivers that comply with UEFI driver model

The early phase DXE drivers depends on presence and contents of a priori file and the evaluation of dependency expression. These early DXE phase drivers contains processor, chipset, and platform initialization code. They also typically include DXE architectural protocols that are required for DXE core to produce EFI System Table, EFI Boot Services, and EFI Runtime Services.

The DXE driver comply with UEFI driver model do not perform any hardware initialization. Instead, they register a Driver Binding Protocol interface in the handle database. The set of Driver Binding Protocols are used by BDS phase to connect the drivers to the devices required to establish consoles and provide access to boot device.