EDK II Topology

S3

All core package code referenced in this document is located in the **GitHub EDK II repository**.

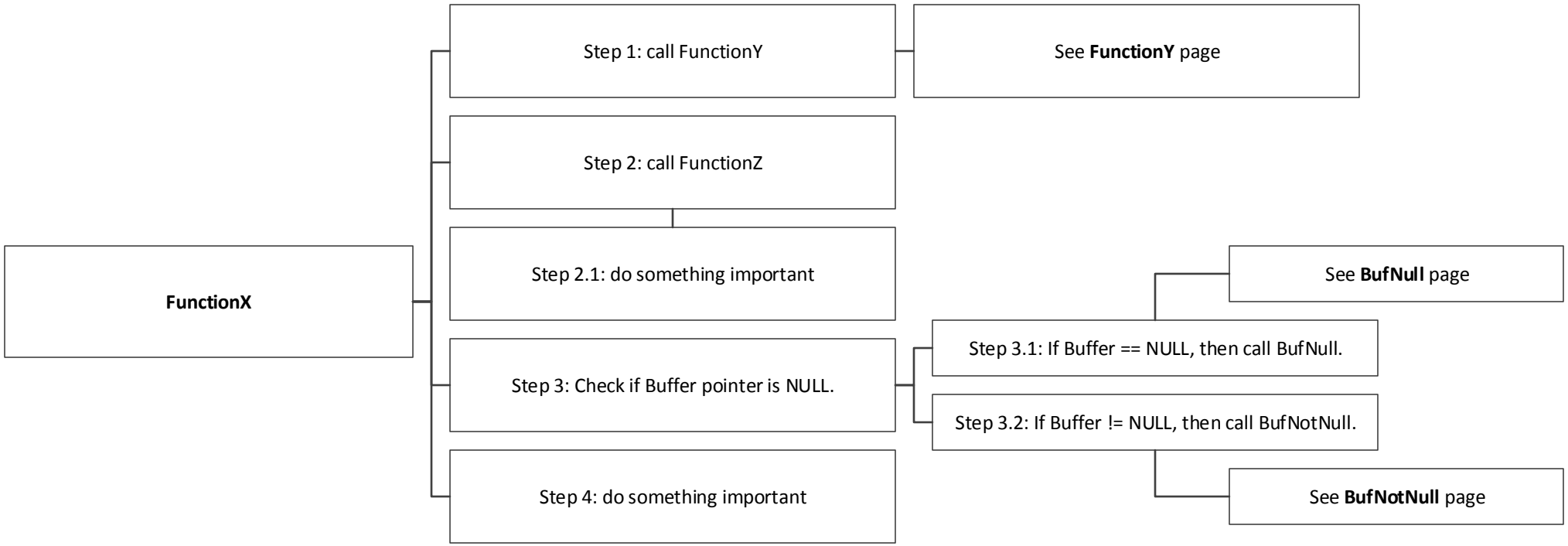
For more in-depth information about EDK II, visit the **Intel® Firmware: Beyond BIOS** page.

Visit TianoCore.org for more **EDK II documentation** and **EDK II projects.**

The function being discussed always starts in the only box on the far left. Boxes represent steps in a function, a branch evaluation in a funtion, or a note to see details on another page about a function being called at a step. Connectors between boxes indicate code flow (who called what) and should be read left-to-right, top-to-bottom. Text in each box will indicate if it's a call, a branch evaluation, or a note. This format was chosen to fit important function details on 1 page.

For the example below, the equivalent C code (note connectors from Step 1-4 to FunctionX):

```
FunctionX() {
        FunctionY();    // see details on FunctionY page
        FunctionZ();    // Step 2; Step 2.1 is in FunctionZ() and is listed because it is important; note the connector between Step 2 and 2.1
        If (Buffer == NULL)     // Step 3
                BufNull();      // Step 3.1
        else
                BufNotNull();   // Step 3.2
        Step 4
}
```



How protocol services are defined so one can find the protocol code to examine it

Summary: if you want to examine the code for a protocol function, you should find the structure definition for the protocol, then find the declaration of the structure, then find the structure member that corresponds with the protocol in the structure definition because they may have different names. EFI_BOOT_SERVICES defines LocateProtocol, mBootServices is of type EFI_BOOT_SERVICES, and the structure member CoreLocateProtocol corresponds with the structure definition LocateProtocol.

MdePkg\Include\UefiSpec.h defines EFI_BOOT_SERVICES structure, and has structure members for protocol services (LocateProtocol, InstallProtocolInterface, etc). MdeModulePkg\Core\Dxe\DxeMain.c has a variable mBootServices of type EFI_BOOT_SERVICES. mBootServices sets function pointers for functions such as LocateProtocol to CoreLocateProtocol and InstallMultipleProtocolInterfaces to CoreInstallMultipleProtocolInterfaces. These functions are defined in MdeModulePkg\Core\Dxe\Handle.c.

How protocols are loaded from flash into memory

Summary: drivers are loaded from flash into memory by some mechanisms into a linked list during the PEI phase.

MdeModulePkg\Core\Dxe\DxeMain.c has DxeMain function which is called when the DXE Core driver is loaded. MdeModulePkg\Core\Dxe\DxeMain.inf has MODULE_TYPE=DXE_CORE and ENTRY_POINT=DxeMain. The end of DxeMain calls CoreInstallMultipleProtocolInterface with the GUID for the HOB that was populated with drivers from the flash part during PEI. PEI phase calls ReadSection (associated with FvReadFileSection in Universal\FirmwareVolume\FwVolDxe\FwVol.c), which eventually gets to a call to LocateProtocol with gEfiDecompressProtocolGuid as a parameter.

Platform DSC libraries and drivers for S3

This is a partial list of libraries that platform DSC file typically uses for S3.

Base libraries
- S3IoLib|MdePkg/Library/BaseS3IoLib/BaseS3IoLib.inf
- S3PciLib|MdePkg/Library/BaseS3PciLib/BaseS3PciLib.inf
- S3SmbusLib|MdePkg/Library/BaseS3SmbusLib/BaseS3SmbusLib.inf
- S3StallLib|MdePkg/Library/BaseS3StallLib/BaseS3StallLib.inf

Null libraries
- S3BootScriptLib|MdePkg/Library/BaseS3BootScriptLibNull/BaseS3BootScriptLibNull.inf

Save State drivers
- MdeModulePkg/Library/PiDxeS3BootScriptLib/DxeS3BootScriptLib.inf
- MdeModulePkg/Universal/Acpi/S3SaveStateDxe/S3SaveStateDxe.inf
- MdeModulePkg/Universal/Acpi/SmmS3SaveState/SmmS3SaveState.inf

S3 Architectural GUIDs

- MdeModulePkg.dec: gEfiAcpiS3ContextGuid
- MdeModulePkg.dec: gFirmwarePerformanceS3PointerGuid
- MdeModulePkg.dec: PcdFirmwarePerformanceDataTableS3Support
- MdeModulePkg.dec: PcdShadowPeimOnS3Boot
- MdePkg.dec: gEfiPeiS3Resume2PpiGuid
- MdePkg.dec: gEfiS3SaveStateProtocolGuid
- MdePkg.dec: gEfiS3SmmSaveStateProtocolGuid

See **PI Volume 5 Chapter 8.6 "PEI Boot Script Executer PPI"** for details.

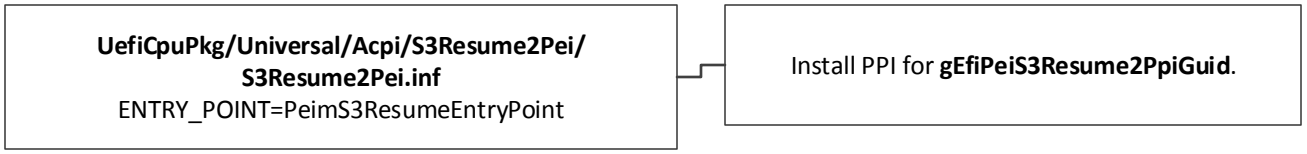
S3 PCDs

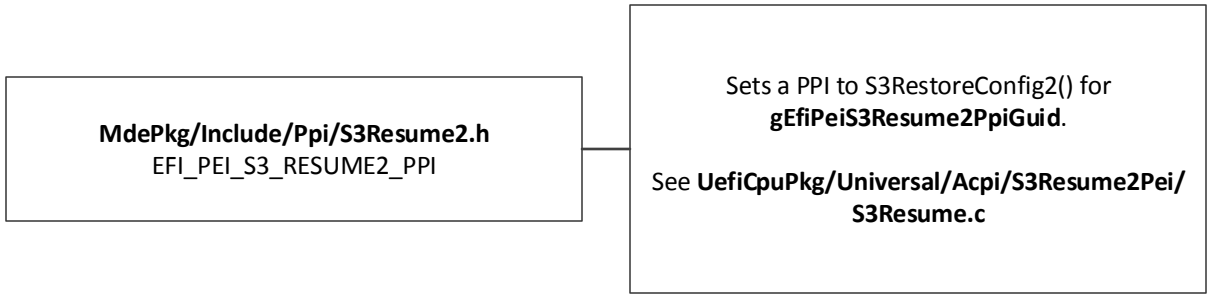
- MdeModulePkg.dec: PcdS3BootScriptTablePrivateDataPtr
- MdeModulePkg.dec: PcdS3BootScriptTablePrivateSmmDataPtr

I/O and Memory to restore on an S3 Resume are saved to a script that is executed on an S3 Resume. The primary functions to save items to the S3 Script:
I/O: S3BootScriptSaveIoWrite(), S3BootScriptSaveIoReadWrite()
Memory: S3BootScriptSaveMemWrite(), S3BootScriptSaveMemReadWrite()
PCI: S3BootScriptSavePciCfgWrite(), S3BootScriptSavePciCfgReadWrite(), S3BootScriptSavePciCfg2Write(), S3BootScriptSavePciCfg2ReadWrite()
See **MdeModulePkg/Library/PiDxeS3BootScriptLib/BootScriptSave.c** for other functions to save items to the S3 Script.

The script memory is allocated when the DxeS3BootScriptLib.inf library constructor is called (see **DxeS3BootScriptLib**). Platform code is responsible for calling the save functions for I/O and memory that are lost across an S3 Save and Resume.

On S3 Resume, S3BootScriptExecute() is called by S3BootScriptExecutorEntryFunction(), which is set to EfiBootScriptExecutorVariable->BootScriptExecutorEntrypoint.
See **MdeModulePkg/Library/PiDxeS3BootScriptLib/BootScriptExecute.c** for details.


UefiCpuPkg/Universal/Acpi/S3Resume2Pei/S3Resume2Pei.inf
ENTRY_POINT=PeimS3ResumeEntryPoint
Install PPI for **gEfiPeiS3Resume2PpiGuid**.



See **PI Volume 5 Chapter 8.6 "PEI Boot Script Executer PPI"** for details.

See **Platform** for steps the platform code must take to initialize S3.

PeiSetBootMode() is called by PEI drivers to set Boot Mode.
MdeModulePkg/Core/Pei/BootMode/BootMode.c

PeiGetBootMode() is called by PEI drivers to get Boot Mode.
MdeModulePkg/Core/Pei/BootMode/BootMode.c

PEI Dispatcher runs.
MdeModulePkg/Core/Pei/Dispatcher/Dispatcher.c

Platform MRC has been dispatched;
Private->PeiMemoryInstalled is set.
PEI Dispatcher checks Boot Mode by comparing
Private->HobList.HandoffInformationTable->BootMode

MdePkg/Include/Pi/PiBootMode.h
BOOT_ON_S3_RESUME

Early PEI before MRC is called, a platform initialization driver detects Boot Mode by reading chipset registers.
Boot Mode is saved by calling PeiServicesSetBootMode().

If Boot Mode is S3, then a PPI might be installed. The PPI GUID can be used to satisfy DEPEX of other drivers.

PeiServicesSetBootMode()
MdePkg/Library/PeiServicesLib/PeiServicesLib.c
This calls SetBootMode(), which is set to PeiSetBootMode() when PEI core is loaded.

This saves Boot Mode to
Private->HobList.HandoffInformationTable->BootMode

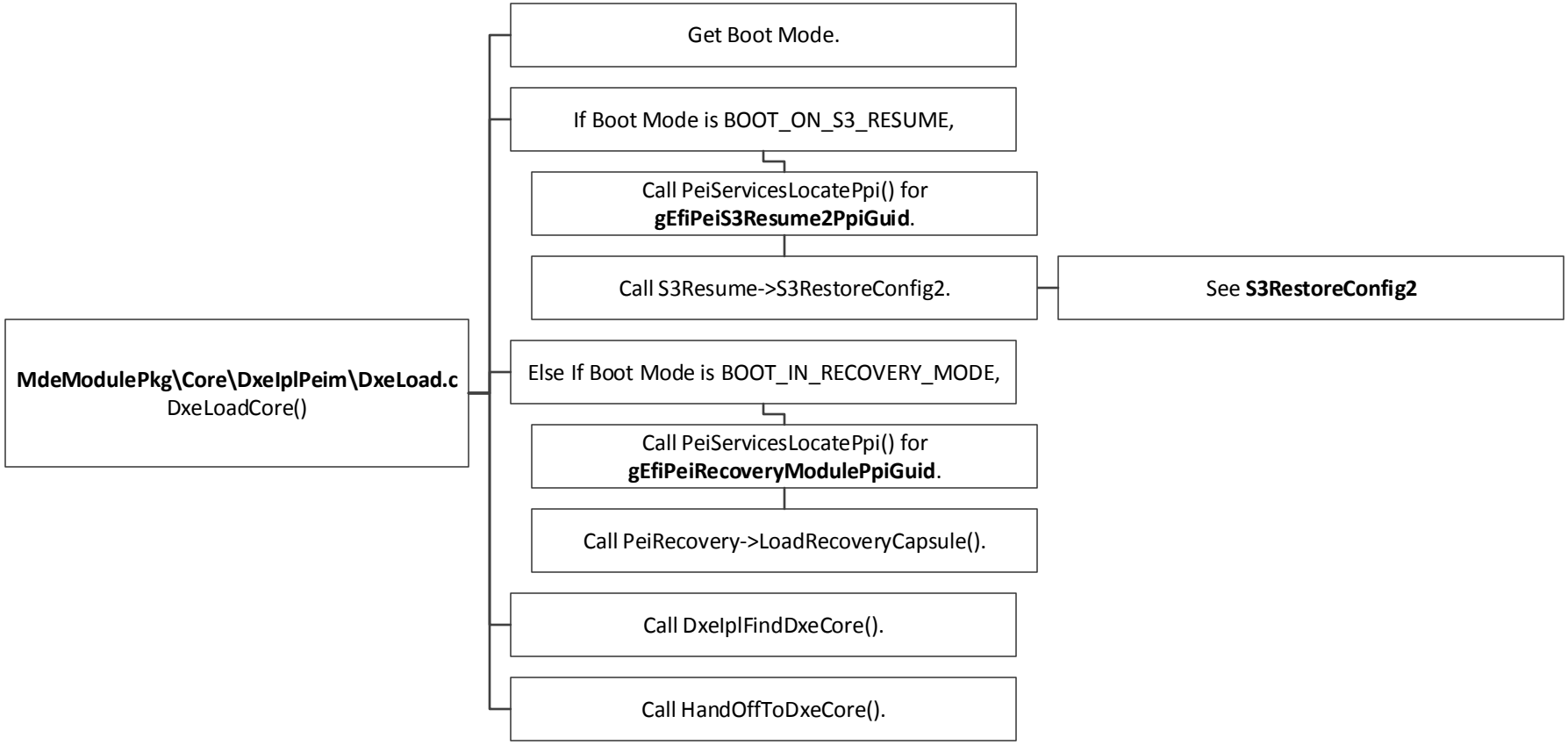
PeiServicesGetBootMode()
This calls GetBootMode(), which is set to PeiGetBootMode() when PEI core is loaded.

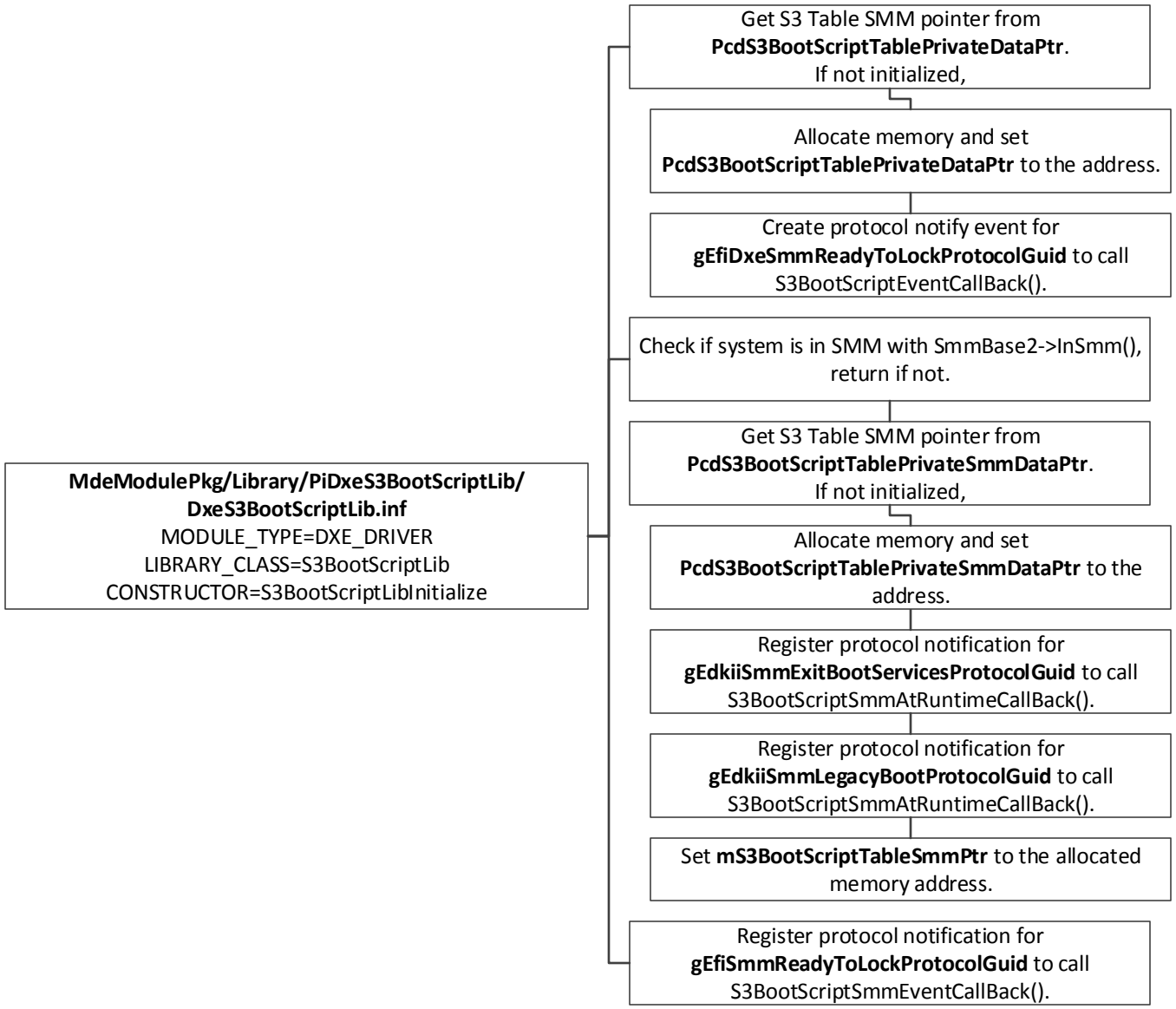
See "EFI_PEI_SERVICES gPs = {"
MdeModulePkg/Core/Pei/PeiMain/PeiMain.c

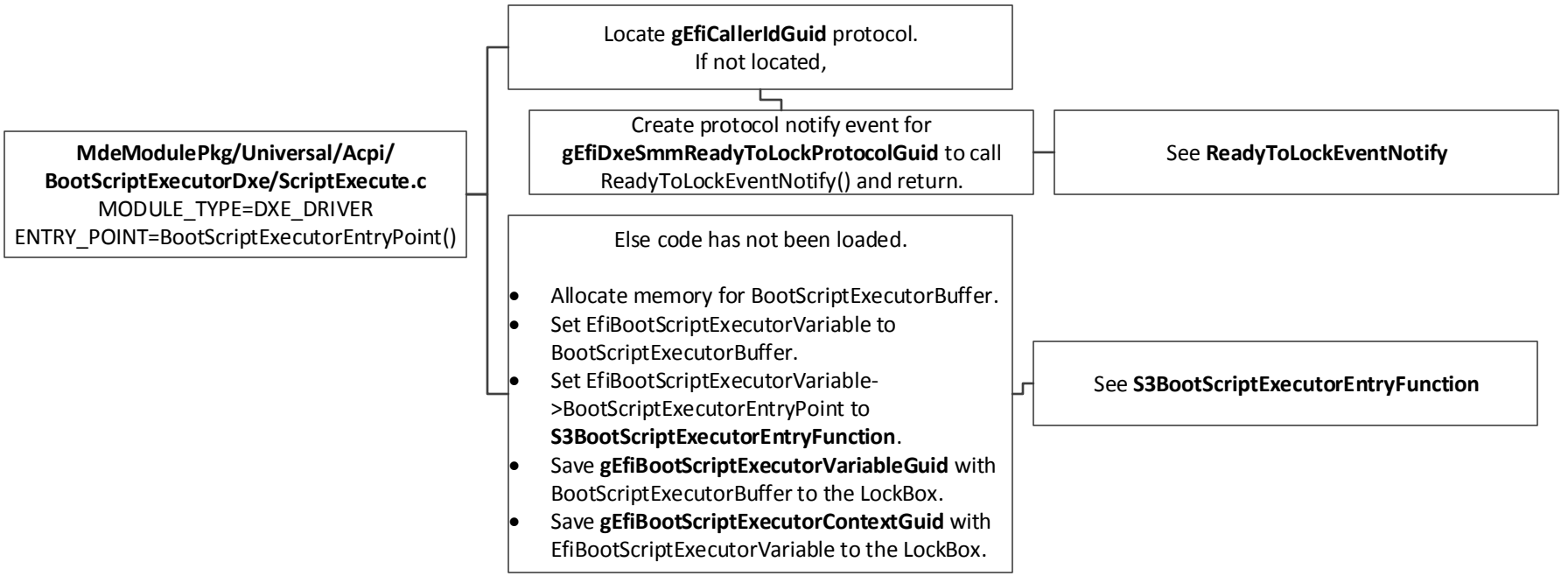
Platform MRC is called; it checks Boot Mode by calling PeiServicesGetBootMode().

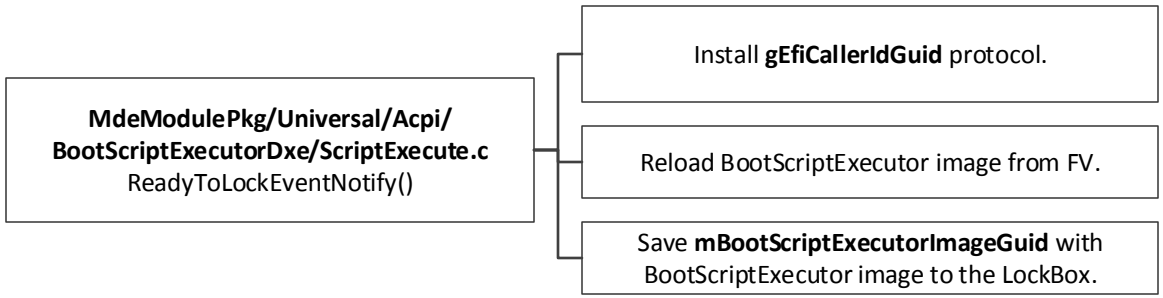
MRC takes different actions based on Boot Mode:
S3 or Flash Update (Capsule)
(BOOT_ON_S3_RESUME, BOOT_ON_FLASH_UPDATE)
No Configuration Changes (fast)
(BOOT_ASSUMING_NO_CONFIGURATION_CHANGES)
Default (cold)

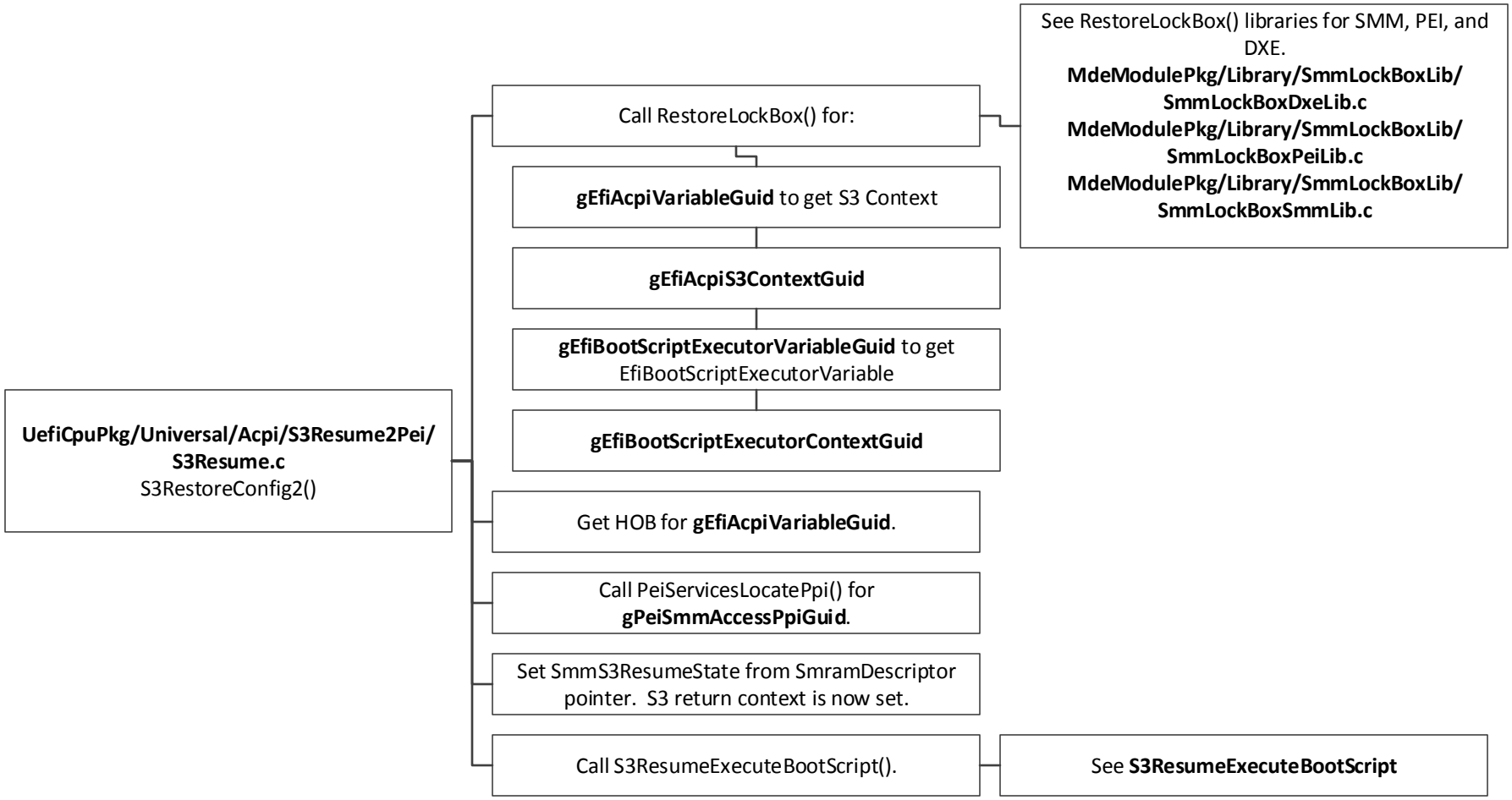
See **PI Volume 1 Chapter 4.3 "Boot Mode Servces"** for details.

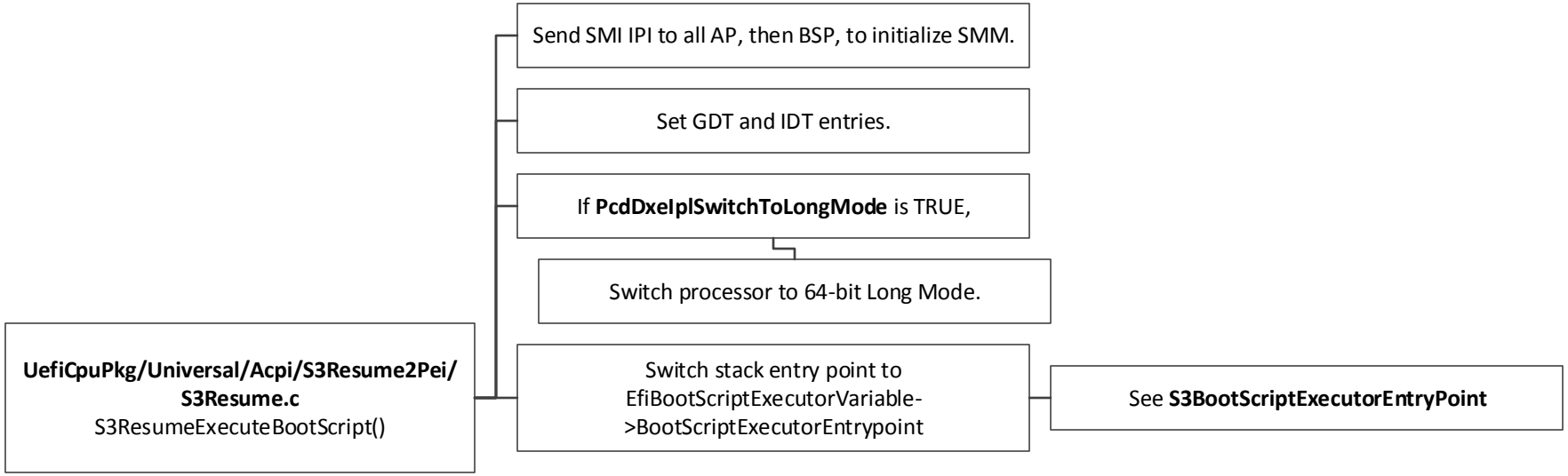

MdeModulePkg\Core\DxeIplPeim\DxeLoad.c
DxeLoadCore()
Get Boot Mode.
If Boot Mode is BOOT_ON_S3_RESUME,
Call PeiServicesLocatePpi() for gEfiPeiS3Resume2PpiGuid.
Call S3Resume->S3RestoreConfig2.
See S3RestoreConfig2
Else If Boot Mode is BOOT_IN_RECOVERY_MODE,
Call PeiServicesLocatePpi() for gEfiPeiRecoveryModulePpiGuid.
Call PeiRecovery->LoadRecoveryCapsule().
Call DxeIplFindDxeCore().
Call HandOffToDxeCore().

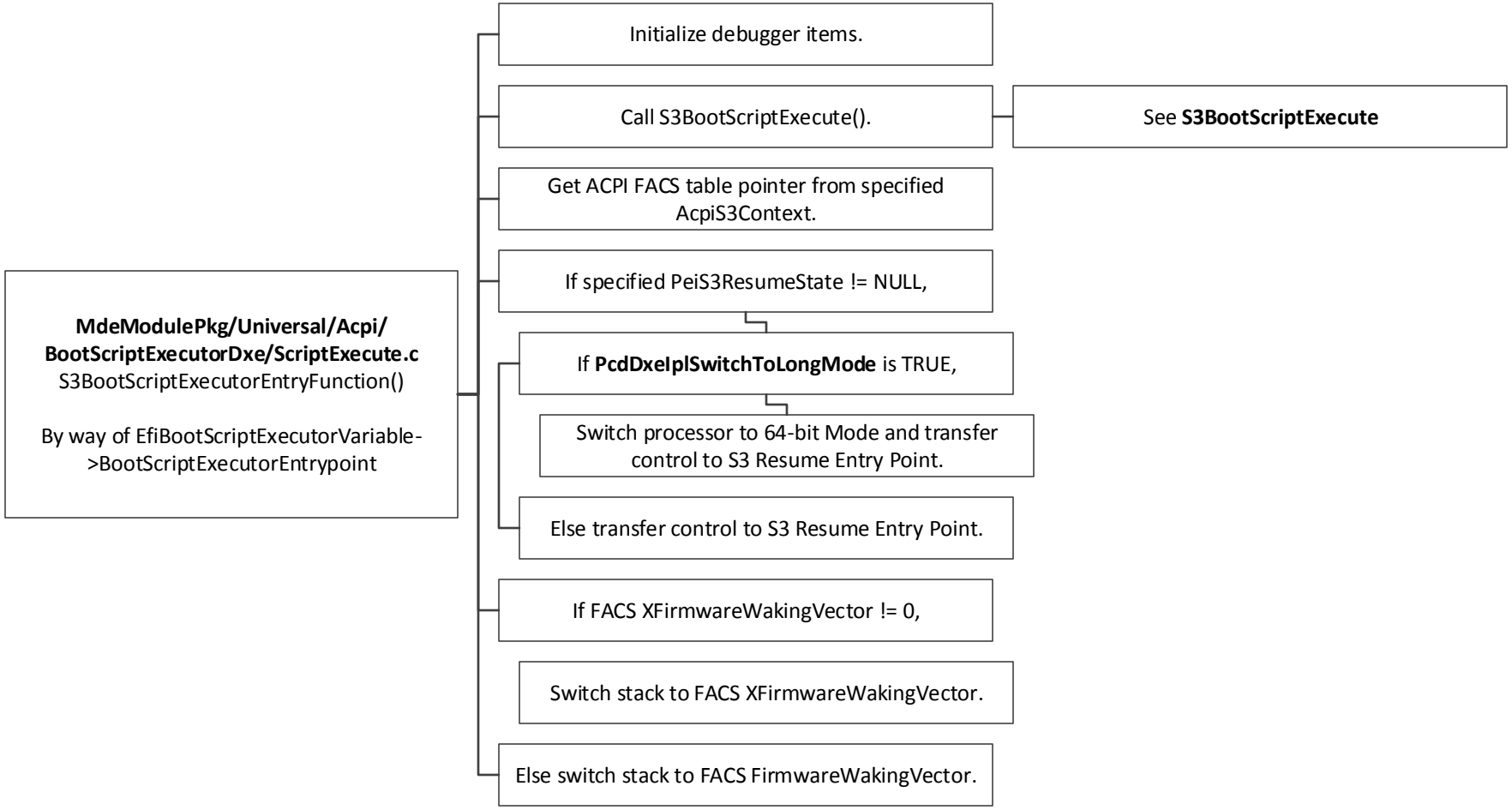

MdeModulePkg/Library/PiDxeS3BootScriptLib/
DxeS3BootScriptLib.inf
MODULE_TYPE=DXE_DRIVER
LIBRARY_CLASS=S3BootScriptLib
CONSTRUCTOR=S3BootScriptLibInitialize
Get S3 Table SMM pointer from
PcdS3BootScriptTablePrivateDataPtr.
If not initialized,
Allocate memory and set
PcdS3BootScriptTablePrivateDataPtr to the address.
Create protocol notify event for
gEfiDxeSmmReadyToLockProtocolGuid to call
S3BootScriptEventCallBack().
Check if system is in SMM with SmmBase2->InSmm(),
return if not.
Get S3 Table SMM pointer from
PcdS3BootScriptTablePrivateSmmDataPtr.
If not initialized,
Allocate memory and set
PcdS3BootScriptTablePrivateSmmDataPtr to the
address.
Register protocol notification for
gEdkiiSmmExitBootServicesProtocolGuid to call
S3BootScriptSmmAtRuntimeCallBack().
Register protocol notification for
gEdkiiSmmLegacyBootProtocolGuid to call
S3BootScriptSmmAtRuntimeCallBack().
Set mS3BootScriptTableSmmPtr to the allocated
memory address.
Register protocol notification for
gEfiSmmReadyToLockProtocolGuid to call
S3BootScriptSmmEventCallBack().

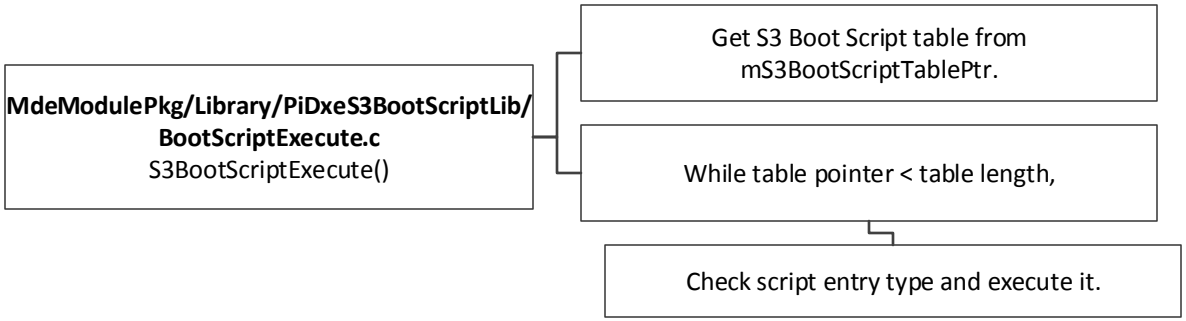

MdeModulePkg/Universal/Acpi/BootScriptExecutorDxe/ScriptExecute.c
MODULE_TYPE=DXE_DRIVER
ENTRY_POINT=BootScriptExecutorEntryPoint()
Locate **gEfiCallerIdGuid** protocol.
If not located,
Create protocol notify event for **gEfiDxeSmmReadyToLockProtocolGuid** to call ReadyToLockEventNotify() and return.
See **ReadyToLockEventNotify**
Else code has not been loaded.
• Allocate memory for BootScriptExecutorBuffer.
• Set EfiBootScriptExecutorVariable to BootScriptExecutorBuffer.
• Set EfiBootScriptExecutorVariable->BootScriptExecutorEntryPoint to **S3BootScriptExecutorEntryFunction**.
• Save **gEfiBootScriptExecutorVariableGuid** with BootScriptExecutorBuffer to the LockBox.
• Save **gEfiBootScriptExecutorContextGuid** with EfiBootScriptExecutorVariable to the LockBox.
See **S3BootScriptExecutorEntryFunction**


MdeModulePkg/Universal/Acpi/
BootScriptExecutorDxe/ScriptExecute.c
ReadyToLockEventNotify()
Install gEfiCallerIdGuid protocol.
Reload BootScriptExecutor image from FV.
Save mBootScriptExecutorImageGuid with
BootScriptExecutor image to the LockBox.


UefiCpuPkg/Universal/Acpi/S3Resume2Pei/S3Resume.c
S3RestoreConfig2()
Call RestoreLockBox() for:
gEfiAcpiVariableGuid to get S3 Context
gEfiAcpiS3ContextGuid
gEfiBootScriptExecutorVariableGuid to get EfiBootScriptExecutorVariable
gEfiBootScriptExecutorContextGuid
See RestoreLockBox() libraries for SMM, PEI, and DXE.
MdeModulePkg/Library/SmmLockBoxLib/SmmLockBoxDxeLib.c
MdeModulePkg/Library/SmmLockBoxLib/SmmLockBoxPeiLib.c
MdeModulePkg/Library/SmmLockBoxLib/SmmLockBoxSmmLib.c
Get HOB for gEfiAcpiVariableGuid.
Call PeiServicesLocatePpi() for gPeiSmmAccessPpiGuid.
Set SmmS3ResumeState from SmramDescriptor pointer. S3 return context is now set.
Call S3ResumeExecuteBootScript().
See S3ResumeExecuteBootScript


Send SMI IPI to all AP, then BSP, to initialize SMM.
Set GDT and IDT entries.
If **PcdDxeIplSwitchToLongMode** is TRUE,
Switch processor to 64-bit Long Mode.
UefiCpuPkg/Universal/Acpi/S3Resume2Pei/S3Resume.c
S3ResumeExecuteBootScript()
Switch stack entry point to EfiBootScriptExecutorVariable->BootScriptExecutorEntrypoint
See **S3BootScriptExecutorEntryPoint**


MdeModulePkg/Universal/Acpi/BootScriptExecutorDxe/ScriptExecute.c
S3BootScriptExecutorEntryFunction()
By way of EfiBootScriptExecutorVariable->BootScriptExecutorEntrypoint
Initialize debugger items.
Call S3BootScriptExecute().
See S3BootScriptExecute
Get ACPI FACS table pointer from specified AcpiS3Context.
If specified PeiS3ResumeState != NULL,
If PcdDxeIplSwitchToLongMode is TRUE,
Switch processor to 64-bit Mode and transfer control to S3 Resume Entry Point.
Else transfer control to S3 Resume Entry Point.
If FACS XFirmwareWakingVector != 0,
Switch stack to FACS XFirmwareWakingVector.
Else switch stack to FACS FirmwareWakingVector.


MdeModulePkg/Library/PiDxeS3BootScriptLib/
BootScriptExecute.c
S3BootScriptExecute()
Get S3 Boot Script table from
mS3BootScriptTablePtr.
While table pointer < table length,
Check script entry type and execute it.