

# Acquiring, Building, and Configuring the Payload Compatible with the Coreboot Reference Bootloader Developed by Intel

Coreboot Reference Bootloader Whitepaper

---

*December 2015*



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: <http://www.intel.com/design/literature.htm>

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2015, Intel Corporation. All rights reserved.



## Contents

---

<b>1.0 Introduction</b> .....	<b>5</b>
1.1 Terminology .....	5
<b>2.0 Payloads</b> .....	<b>6</b>
2.1 UEFI Payload.....	6
2.2 U-Boot.....	6
2.3 SeaBIOS.....	6
<b>3.0 Building Payloads</b> .....	<b>8</b>
3.1 Building UEFI Payload for Coreboot.....	8
3.2 Building U-Boot Payload for Coreboot .....	9
3.3 Building SeaBIOS Payload for Coreboot.....	9
<b>4.0 Conclusion</b> .....	<b>11</b>

## Figures

Figure 1. Coreboot with Different Payloads.....	7
---	---

## Tables

Table 1. Terminology .....	5
----------------------------	---



## Revision History

---

Date	Revision	Description
December 2015	001	Initial release.

§



## 1.0 Introduction

---

This paper provides instructions on how to acquire, build and configure the payloads that are compatible with the Coreboot reference bootloader that is developed by Intel®. The payloads supported and described in this document cover boot support for the most common Operating Systems for Internet of Things Group (IOTG) customers.

### 1.1 Terminology

Table 1. Terminology

Term	Description
BIOS	Basic Input/Output
EFI	Extensible Firmware Interface
Intel® FSP	Intel® Firmware Support Package
IOTG	Internet of Things Group
UEFI	Unified Extensible Firmware Interface



## 2.0 Payloads

---

While the Coreboot is meant to provide minimal board level initialization, the Intel® Firmware Support Package (Intel® FSP) can be used to provide Silicon Initialization. Coreboot will hand off control to a Payload which can then perform more initialization and ultimately hand off control to the Operating System (OS). [Figure 1](#) shows how different payloads can be incorporated into the Coreboot bootloader. These differing payloads may be included into the Coreboot in order to boot various OS or provide differing bootloader functionality.

### 2.1 UEFI Payload

The Unified Extensible Firmware Interface (UEFI) Payload is based on the Open Source Tianocore firmware. As a payload, it provides an Extensible Firmware Interface (EFI) shell to Coreboot and allows for booting the UEFI OS. The UEFI payloads can be implemented on a variety of computer architectures, and even on some embedded systems. This is one of the more common payloads supported and found on platforms. Most Intel Coreboot reference bootloaders use the UEFI Payload to boot to Microsoft\* Windows\*, Android\*, and Yocto\* OSs.

### 2.2 U-Boot

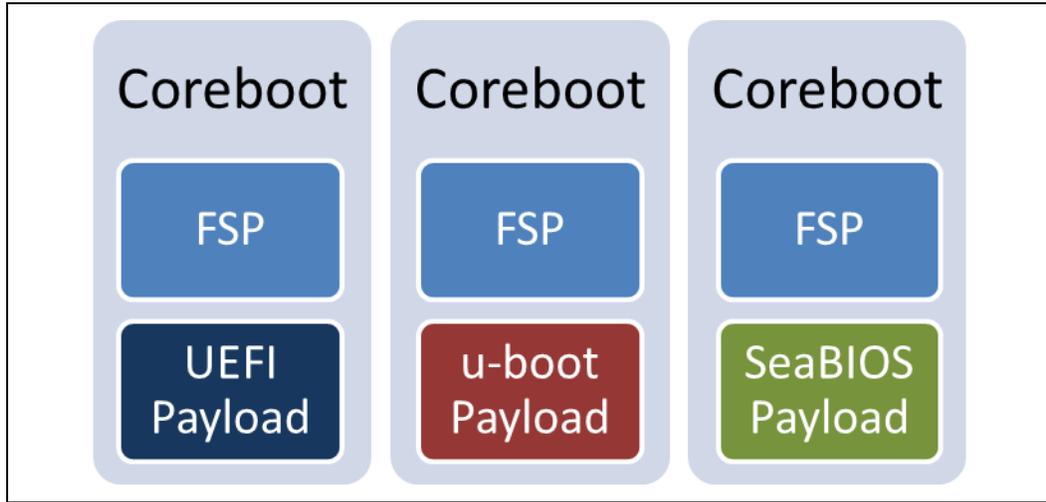
U-Boot is an open source bootloader intended to boot to a devices OS kernel. The U-Boot bootloader is used on a variety of different embedded devices to bring up the devices OS. This bootloader is applicable to a variety of different computer architectures including ARM, x86 and so on. The Intel Coreboot reference bootloader supports booting to Linux\* though the U-Boot payload.

### 2.3 SeaBIOS

The SeaBIOS payload is open source and available for x86 computer architectures. SeaBIOS was built off the open source Basic Input/Output System (BIOS) implementation of Bochs emulator. SeaBIOS payload is unique as it provides a 16-bit x86 BIOS implementation. This payload can be run on bare hardware as a Coreboot payload or on emulators. The Intel Coreboot reference bootloader uses SeaBIOS to boot to Windows\* 7 and older versions of Windows\*.



Figure 1. Coreboot with Different Payloads





## 3.0 Building Payloads

---

This section provides step by step instructions on how to build the three different payloads for Coreboot.

### 3.1 Building UEFI Payload for Coreboot

Go to <http://firmware.intel.com/develop> and under the coreboot\*, coreboot Module/**Payload** dropdown consists of the following:

- **2014-WW26-UEFI.CoreBoot.Payload – ReleaseNotes.txt**
- **CorebootModulePkg.Zip**

This sip file contains the source code package of Coreboot Support Modules that will be used to parse the coreboot tables in memory, report memory/IO resources and install acpi and smbios tables from Coreboot into EFI system tables.

- **CorebootPayloadPkg.zip**

This zip file contains the source code package of Coreboot Payload Modules, provides definitions of payload image's layout, and lists the module required in DSC file.

1. The encryption code for **.zip** file can be found in the text file a couple paragraph's below (**2014-WW26-UEFI.coreboot.payload.zip-DecryptionKey.txt**) on <http://firmware.intel.com/develop>
2. Unzip **2014-WW26-UEFI.coreboot.coreboot.payload.zip** into your working directory.
3. Go to your working directory and unzip the **CorebootModulePkg.zip** and **CorebootPayload.zip** which was contained in the zip file unzipped in the previous Step.
4. Inside the **corebootPayloadPkg** folder, open the **BuildAndIntegrationInstructions.txt** and find out which revision number you will need for **EDK II** code. For **2014-WW26-UEFI.coreboot.payload.zip** use revision **15472**.
5. Download the **EDK II** repository with the revision found in previous instruction from <http://svn.code.sf.net/p/edk2/code/branches/UDK2014/> by using SVN, this will be the working directory.
6. Copy all unzipped files from **\*.coreboot.payload.zip** into the **EDK II** directory.
7. Run **edksetup.bat**. (See Note Below)
8. Run any of the following commands: (for clean build, run "**build clean**")
  - For the debug ia32 build:

```
build -a IA32 -p CorebootPayloadPkg\CorebootPayloadPkgIA32.dsc -b  
DEBUG -t <ToolChain>
```



- For release **ia32 build**:  
 build -a IA32 -p CorebootPayloadPkg\CorebootPayloadPkgIA32.dsc -b  
**RELEASE -t <ToolChain>**
  - For debug **X64 build**:  
 build -a IA32 -a X64 -p CorebootPayloadPkg\CorebootPayloadPkgX64.dsc -b  
**DEBUG -t <ToolChain>**
  - For release **X64 build**:  
 build -a IA32 -a X64 -p CorebootPayloadPkg\CorebootPayloadPkgX64.dsc -b  
**RELEASE -t <ToolChain>**
9. Payload image (**UEFIPAYLOAD.fd**) will be generated inside the folder  
 Build\CorebootPayloadPkg.
  10. Copy payload image (**UEFIPAYLOAD.fd**) to coreboot workspace  
 /payload/external/UEFI/.
  11. Use "**make menuconfig**" and change payload name/path.

**Note:** The script might not setup the toolchain correctly, you might need to change file  
 Conf\tools\_def.txt to identify the location of the toolchain.

For example, added the following in green:

```

DEFINE VS2008_BIN      = C:\Program Files (x86)\Microsoft
Visual Studio 9.0\vc\bin

DEFINE VS2008_DLL      = C:\Program Files (x86)\Microsoft
Visual Studio 9.0\Common7\IDE;DEF(VS2008_BIN)

DEFINE VS2008_BINX64   = DEF(VS2008_BIN)\x86_amd64

DEFINE VS2008_BIN64    = DEF(VS2008_BIN)\x86_ia64
  
```

## 3.2 Building U-Boot Payload for Coreboot

1. Get U-Boot repository:  
 git clone <http://git.denx.de/u-boot.git>
2. CD to the **U-Boot** directory.
3. make **distclean**
4. make **coreboot-x86**
5. Payload image (U-Boot) will be generated
6. Copy Payload (U-Boot) to the Coreboot workspace/payload/external/**uboot**.
7. Use "**make menuconfig**" and change payload name/path.

## 3.3 Building SeaBIOS Payload for Coreboot

1. From the Coreboot top level directory, go to the Coreboot configuration by typing:  
 "**make menuconfig**"



2. Select **SeaBIOS** as payload by going to Payload → Add a payload → **SeaBIOS**
3. Exit the Coreboot configuration and save the changes.
4. Build the Coreboot by typing "**make**".

The SeaBIOS will automatically be downloaded, built, and integrated into the Coreboot

§



## 4.0 Conclusion

---

With the help of open source software, such as Coreboot, we are able to provide different payloads. These payloads are essential when developing and maintaining board functionality. We hope you found this document helpful in your development. To obtain support from the Coreboot community, subscribe to the Coreboot mailing list ([www.coreboot.org/Mailinglist](http://www.coreboot.org/Mailinglist)) or read the FAQ ([www.coreboot.org/FAQ](http://www.coreboot.org/FAQ)) for details. If you would like to send feedback or ask a questions, submit them to [www.intel.com](http://www.intel.com) and send us your feedback.

§