# EDK2 Duet payload for coreboot

EDK2 Duet is project that packages the DXE portion of UEFI as a self-contained module. This allows legacy BIOS or coreboot firmware to handle chipset and memory initialization before passing control to UEFI DXE code for boot device initialization and OS launch. Duet can run only after a legacy BIOS or coreboot firmware has completed chipset and memory initialization. This patch set is tested with a recent revision (14928) of the [EDK2](https://sourceforge.net/apps/mediawiki/tianocore/index.php?title=EDK2) Duet project. Major changes are:

* Support running as a coreboot payload
* Fix known problems and confirm proper operation on real hardware
* Simplify and document build procedure for both Windows and Linux

The build procedure creates two files: EFILDR16 and FAT16.img. EFILDR16 is a coreboot payload. FAT16.img is a legacy bootable disk image. The files are in edk2\Build\images.

At this time, coreboot payload operation has been tested on ASRock E350M1 only. With coreboot+duet payload running on this board, OS installation and booting works properly with all operating systems tested: Ubuntu 13.10, Windows 7, Windows 8.1.

Starting Duet from a bootable disk image is useful for code development and debug. Legacy BIOS bootable disk images can be easily booted by emulators such as QEMU or AMD Simnow. The bootable disk image has been tested with many simnow models, as well as the following real hardware: ASRock E350M1, Asus Sabertooth P67 + AMD HD 7870, HP Pavilion DV4, HP Pavilion DM1, Asus A75M-HVS.

# Limitations:

* Only UEFI boot mode is supported. Legacy boot is not supported. Duet will not boot DOS or Windows XP for example. Windows Vista and newer support UEFI mode boot. Major Linux distributions support UEFI boot mode and will work with Duet.
* Legacy option rom support is limited to video only. Because Duet AHCI support doesn’t use an option rom, this limitation doesn’t prevent ACHI mode boot.
* Only the x64 build target is supported. While the standard Duet project also allows a 32-bit build, 32-bit UEFI is not in widespread use today. This patched Duet boots only 64-bit, UEFI aware operating systems. 32-bit boot support could be added back if needed.
* On AMD systems, a USB keyboard doesn't work before OS startup. This is because EDK2 does not support USB OCHI, the controller AMD uses for USB 1.x. Work-around: for AMD systems, run the USB keyboard through a 2.0 hub.
* Booting from a USB 3.0 device doesn’t work. Tested on Asus P67 only.
* UEFI NVRAM support is work in progress, and will only be available for Duet coreboot payload mode. A fake NVRAM system adapted from the OVMF project is used when flash memory based NVRAM is not available. Known limitations of fake NVRAM:
  + If multiple operating systems are installed, the boot order will not be maintained.
  + After a power down causes loss of fake NVRAM, Ubuntu Linux will not boot. The built-in UEFI shell boots instead. Work-around: copy EFI\ubuntu\shimx64.efi to EFI\BOOT\BOOTX64.EFI.

# Tool Setup and Building: Windows

Building this version of Duet from Windows requires Microsoft Visual Studio. Versions 2008, 2010 and 2012 have been tested and versions 2003 and 2005 may also work. The build process will locate the Visual Studio install directory automatically. Also required is JWASM.exe (version 2.11). Place JWASM.exe in the BaseTools\Bin\Win32 directory. Windows 7 x64 was used for testing the build setup. Other x64 editions of Windows will probably work. 32-bit versions of Windows might work. To start the Windows build, run build-windows.bat. EFILDR16 and FAT16.img can be found in edk2\Build\images when the build completes.

# Tool Setup and Building: Linux

Building from Linux was tested using a fresh install of ubuntu-13.10-desktop-amd64.iso. The included gcc 4.8.1 compiler is used. Also required is jwasm (version 2.11). Place jwasm in the BaseTools/Bin/linux directory. To start the build, run build-linux.sh. EFILDR16 and FAT16.img will be placed in edk2/Build/images when the build completes.

# Build Options

Some useful build options are included inside the build batch/script files:

BUILDTARGET: Set according to comments. Use BUILDTARGET=RELEASE once development is complete.

DEBUG\_PRINT\_LEVEL: Set according to comments. Use DEBUG\_PRINT\_LEVEL=0x80000000 once development is complete.

TOOL\_CHAIN\_TAG: For Windows, match this to the version of Visual Studio you have installed. For Linux, this only controls compile flags. The gcc from /usr/bin will be used. For Linux, just leave TOOL\_CHAIN\_TAG=GCC47 and the compile flags will be suitable for use with gcc 4.8.1.

# Archive Contents

* **coreboot\** Modified coreboot snapshot. Changes include payload support for Duet, ACPI additions needed for Windows recognition of PS/2 keyboard and serial port, and temporary work-arounds for Windows build problems.
* **edk2\** edk2 revision 14928 with patches applied. Includes SVN metadata.
  + **build-linux.sh** script file for building from Linux.
  + **build-windows.bat** batch file for building from Windows.
  + **buildlog-linux.txt** Output from Linux build.
  + **buildlog-windows.txt** Output from Windows build.
* **edk2**-patches edk2 changes in patch form.
* **msys** files needed for building coreboot in Windows.
* **coreboot-25dd247.7z** Unmodified coreboot snapshot.
* **apply-edk2-patches.bat** batch file for applying EDK2 patches.
* **build-env-coreboot.bat** batch file for setting up coreboot build environment in Windows.
* **release-notes.docx** This document.
* **bootlog-ubuntu.txt** serial output from coreboot-duet-Ubuntu boot. Coreboot logging is disabled.

# Duet Patch descriptions

**jwasm-windows**: Use JWASM in place of the Microsoft assembler for non-custom build modules (DuetPkg/BootSector is custom build and its assembler selection is done by its makefile). Because JWASM is available for both Windows and Linux, it has the potential to enable asm code sharing for the Linux and Windows builds. JWASM also eliminates the need to obtain a 16-bit Microsoft linker. This patch switches the Windows build from Microsoft MASM to JWASM. The DuetPkg/BootSector MASM to JWASM switch is handled by a separate patch. After applying this patch, download JWASM.exe and copy it to BaseTools\Bin\Win32.

**build-windows**: Use the standard environment variable method to locate Visual Studio installations to remove the hard-coded assumption of a C drive installation. Add a batch file to automate the build. Supports Visual Studio 2003, 2005, 2008, 2010, and 2012. Tested with Visual Studio 2008, 2010 and 2012.

**build-bootsector:** Build the BootSector binary files from asm source so that they can be modified. Switch both Windows and Linux builds to JWASM for this module so that a single source file can be used for both. Sync the Linux and Windows makefiles for easier diffing. After applying this patch, download JWASM.exe and copy it to BaseTools\Bin\Win32. Download jwasm for Linux and copy it to BaseTools/Bin/linux.

**rtc-amd**: Fix RTC initialization error on AMD systems. PcRtc.c wants to see register D bit 7 (VRT) high almost immediately after writing the below value, which clears it with the default UEFI value of zero. The AMD SB700 updates this bit only once per 4-1020ms (1020ms default). This causes function RtcWaitToUpdate to return an error. Preset VRT to 1 to avoid this.

**legacybiosthunk-safe-int15h**: Don't call INT 15h if no INT15h vector is installed. When coreboot runs Duet as a payload, there is no legacy BIOS and no INT 15h. In this case,

coreboot is responsible for unmasking A20.

**legacybiosthunk-logic**: Remove if statements where a logic error causes them to always evaluate to true.

**legacybiosthunk-initflags**: Clear the eflags value that will be popped when the legacy BIOS code executes IRET to return control to the caller. It is important that

trap flag is not set in this value because the resulting INT1 will stop execution.

**buildfatimage**: Add a utility to build a legacy bootable FAT image containing Duet UEFI. Though the logical drive capacity is 32MB, truncation of unused trailing clusters keeps the file size small. This image can be raw copied to a USB flash drive for testing on real hardware. AMD Simnow can also boot the image.

**debug-fixes**: Fix problem where uninitialized cursor position data causes overwrite of memory at address 0 if function ClearScreen() is not called. Remove some console debug prints when the logging level is set for error only. Filter non-displayable character code (\r) before writing to the video buffer.

**paging-fixes**: Correct the count calculation when creating 4KB pages for the lower 2MB of memory. Ensure adequate memory is available before building page tables.

**efiloader**: Fix problem with debug message where incorrect format type causes every other character to be omitted. Add defines for removing hard-coded memory address.

**bdsplatform**: Correct print level informational debug messages. Reduce splash screen display time. Remove emulation work-around.

**pcidevicesupport**: Correct print level informational debug message.

**pcatpcirootbridge**: Correct print level informational debug message.

**satacontroller**: Suppress error message for ‘sata controller already started’ because this is apparently normal.

**idemode**: Reduce 35 second wait because IDE devices should respond more quickly than this. This prevents a malfunctioning IDE device from adding 35 seconds to the boot time. Correct print level informational debug message.

**page**: Correct print level informational debug message.

**terminalconout**: correct cursor control escape sequence generation when the column number is greater than 99.

**partition**: Prevent trying to process a partition boot record as if it were a master boot record.

**pcd-service**: Fix PCD problem: <https://sourceforge.net/apps/trac/edk2/ticket/596>.

**8259legacymask**: Remove hard-coded setting to enable legacy interrupt mask bit 0 (8254 periodic interrupt). The mask bits can be controller at build time using Pcd8259LegacyModeMask or at run time by calling Interrupt8259SetMode().

**cpugdt**: Fix pointer truncation problem that prevents booting when more than 4GB of memory is installed.

**dxesmmcpuexception**: Work around a Duet problem where an unexpected interrupt halts the system. The unexpected interrupt occurs during PS/2 keyboard initialization (Ps2KeyboardDxe) on some systems using AMI Aptio UEFI BIOS. The unexpected interrupt occurs because the Aptio SMM code unmasks PS/2 keyboard interrupts.

**setcodeselector**: Fix pointer truncation problem that prevents booting when more than 4GB of memory is installed.

**spurious**: Prevent spurious interrupts from leading to a boot failure. Spurious interrupts sometimes occur on real hardware when the 8259 interrupt controller is used. Duet can easily generate a spurious interrupt because it reprograms the 8254 after it has already been start by the legacy BIOS.

**e820**: For coreboot payload operation, use code from SeaBIOS to generate the needed e820 data.

**fake-nvram**: Remove Duet NVRAM solution and replace with one adapted from the OVMF EmuVariableFvbRuntimeDxe project.

**uefi-memory-map**: Move PrepareHobCpu() so that global data variable is updated before use. Add argument to function PrepareHobPhit() for passing memory size consumed by page tables so that hard coding can be removed. Remove code that reports DxeCore memory as free. Otherwise, gEventQueue freed and reused while gEventQueue is still in operation, leading to a crash. Remove some unused static structures.

**use-std-metronome**: Replace the Duet LegacyMetronome module with the standard Metronome module. The standard module works well and there is no obvious reason for replacing it.

**dsc-fdf**: Customize the DSC and FDF build files: Add XCHI, remove PS/2 floppy, add network boot, add optional logo, build shell from source code.

**misc**: Remove some hard-code constants and replace with defined symbols. Miscellaneous changes for coreboot payload use and for debugging.

# Coreboot patch descriptions

Patches for coreboot have not be made yet. The changes are small and can be seen by comparing the included coreboot directory to the coreboot snapshot (coreboot-25dd247.7z). Only the payload and ACPI related changes are significant. Other changes are temporary work-arounds for Windows build problems.