# 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 other code 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 chipset and memory initialization has completed. This patch set is tested with SVN revision 14981 (12/13/2013) 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 can be found in edk2\Build\images at build completion.

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.

# Download

12/06/2013 [e350m1-duetpayload-032.7z](http://notabs.org/coreboot/duet-payload/e350m1-duetpayload-032.7z) initial version.

12/13/2013 [e350m1-duetpayload-033.7z](http://notabs.org/coreboot/duet-payload/e350m1-duetpayload-033.7z):

* Adapt the build environment so that no coreboot modifications are needed.
* Update the patched EDK2 to SVN revision 14981 (12/13/2013) and rebase the patches to the same.
* Update the included coreboot to [revision](http://review.coreboot.org/gitweb?p=coreboot.git;a=shortlog) f0a13ceb639f7a7d5a6e84a2c89f3deab0de757a (12/11/2103).

# 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 USB 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 be available only 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 loss of fake NVRAM due to power down, Ubuntu Linux will not boot. The built-in UEFI shell boots instead. Work-around: copy EFI system partition file EFI\ubuntu\shimx64.efi to EFI\BOOT\BOOTX64.EFI.
* With the included unmodified coreboot snapshot, Windows will not recognize a PS/2 keboard. Work-around: use superio.asl from the [initial version](http://notabs.org/coreboot/duet-payload/e350m1-duetpayload-032.7z) of this archive.

# Tool Setup for Building Duet: 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 (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 for Building Duet: Linux

Starting with a clean install of Ubuntu-13.10-desktop-amd64.iso: Download jwasm (version 2.11) and place the BaseTools/Bin/linux directory. Install these packages: sudo apt-get uuid-dev iasl g++. To start the build, run build-linux.sh. EFILDR16 and FAT16.img will be placed in edk2/Build/images when the build completes.

# Tool Setup for Building coreboot: Windows

The msys directory of the archive contains all the tools needed to build coreboot from Windows. After extracting the archive, run build-env-coreboot.bat. Now run the make command. When prompted to answer coreboot configuration questions, press Enter to choose the default answer for all questions except for the following:

* Mainboard vendor: **11. ASROCK**
* Mainboard model: **2. E350M1**
* Add a VGA BIOS image: **Yes**
  + VGA BIOS path and filename: **../optionroms/vga.rom**
* Add a payload: **1. None**

Now run the make command again. When the build will completes a summary is printed:

Name Offset Type Size

cmos\_layout.bin 0x0 cmos\_layout 1776

pci1002,9802.rom 0x740 optionrom 57856

fallback/romstage 0xe980 stage 345540

fallback/coreboot\_ram 0x62f80 stage 206083

config 0x954c0 raw 3888

(empty) 0x96440 null 3577624

Now run this command to add the Duet payload:

**build/cbfstool build/coreboot.rom add-flat-binary -f ../edk2/build/images/EFILDR16 -n fallback/payload -l 0x100000 -e 0x102000**

Flash the Asrock E350 board using file build/coreboot.rom.

# Tool Setup for Building coreboot: Linux

Make the cross compile tools using util/crossgcc/Makefile. Then follow the Windows build instructions above.

# Duet 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 Ubuntu 13.10, just leave TOOL\_CHAIN\_TAG=GCC47 and the compile flags will be suitable for use with gcc 4.8.1.

# Archive Contents

* **coreboot\** The unmodified coreboot snapshot used for building the included binaries and testing on real hardware.
* **edk2\** Edk2 revision 14968 with patches applied. Includes SVN metadata for easy syncing with EDK2 updates.
  + **build-linux.sh** Script file for building Duet from Linux.
  + **build-windows.bat** Batch file for building Duet from Windows.
  + **buildlog-linux.txt** Sample output from Linux Duet build.
  + **buildlog-windows.txt** Sample output from Windows Duet build.
* **edk2-patches** Edk2 changes in patch form.
* **msys** Files needed for building coreboot from Windows.
* **coreboot-f0a13ce.7z** Archive containing 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 uses custom build and its assembler selection is done by its makefile). Because JWASM is available for both Windows and Linux, it can make possible asm code sharing between 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 and 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 southbridges (SB700 for example) update 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**: To avoid a crash, 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**: Improve code readability by removing 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 controlled at build time using Pcd8259LegacyModeMask or at run time by calling Interrupt8259SetMode(). This change allows Duet to run legacy video option ROM code with the timer interrupt initially masked for maximum reliability.

**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. The SeaBIOS code is modified to accommodate the Microsoft C compiler’s lack of C99 support.

**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 memory is 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 clear 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-coded constants and replace with defined symbols. Miscellaneous changes for coreboot payload use and for debugging.

**Delete-unused:** Remove some unused files.