presented by

intel®

UEFI Community Resources

UEFI Spring Plugfest – May 8-10, 2012
Presented by Brian Richardson,
Intel Corporation

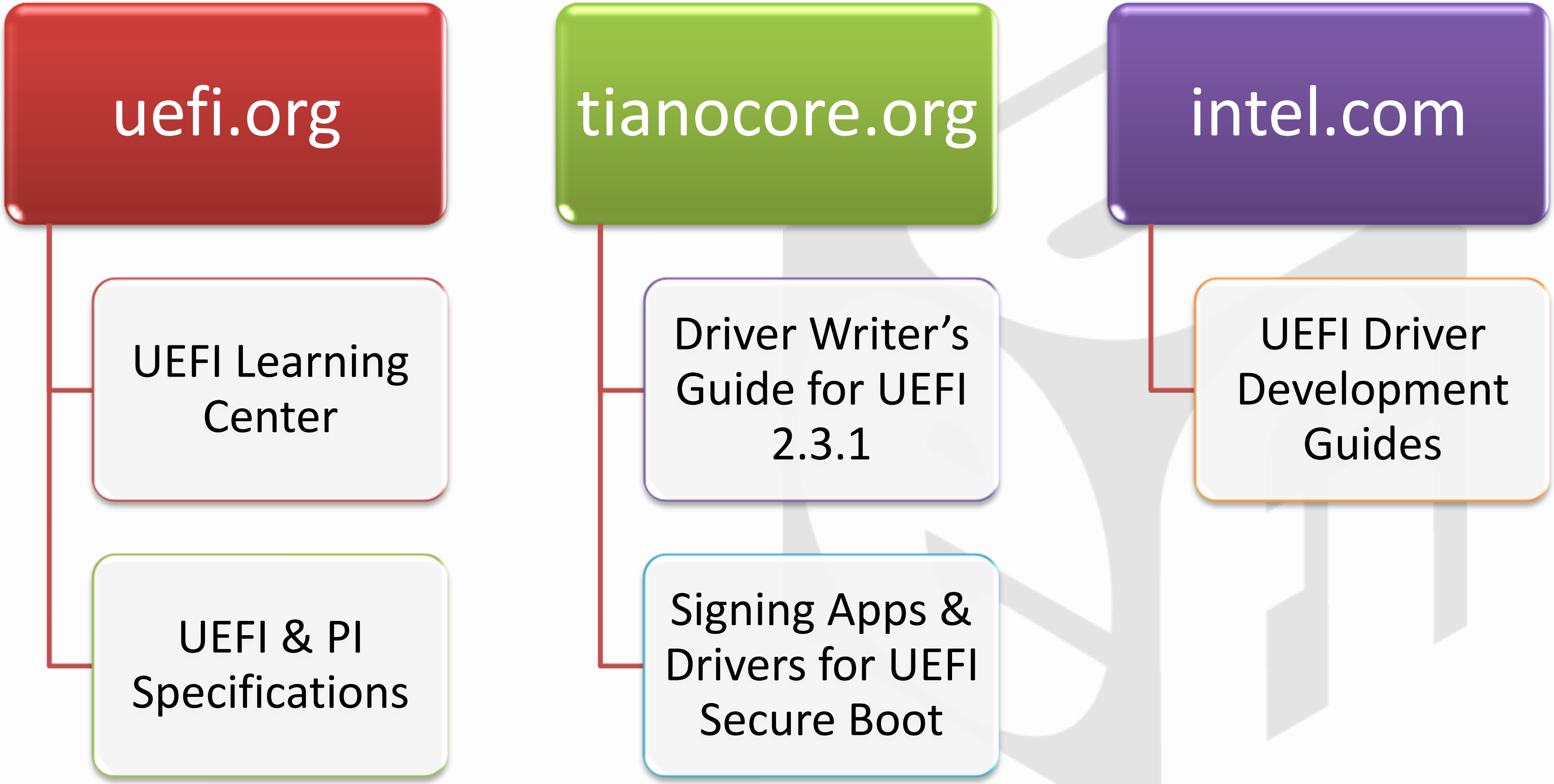
Agenda

- Exploring the UEFI Resources
- Documentation Resources
- Development Resources
- The Intel UEFI Community Resource Center
- Summary / Q&A

Exploring UEFI Resources

- UEFI Has a Robust Developer Community
  - Documentation Resources
  - Development Tools
  - Based on Open Source Projects & Member Company Contributions
- Developers need to check several locations to see all of these resources

Documentation Resources



uefi.org

- UEFI Learning Center
- UEFI & PI Specifications

tianocore.org

- Driver Writer's Guide for UEFI 2.3.1
- Signing Apps & Drivers for UEFI Secure Boot

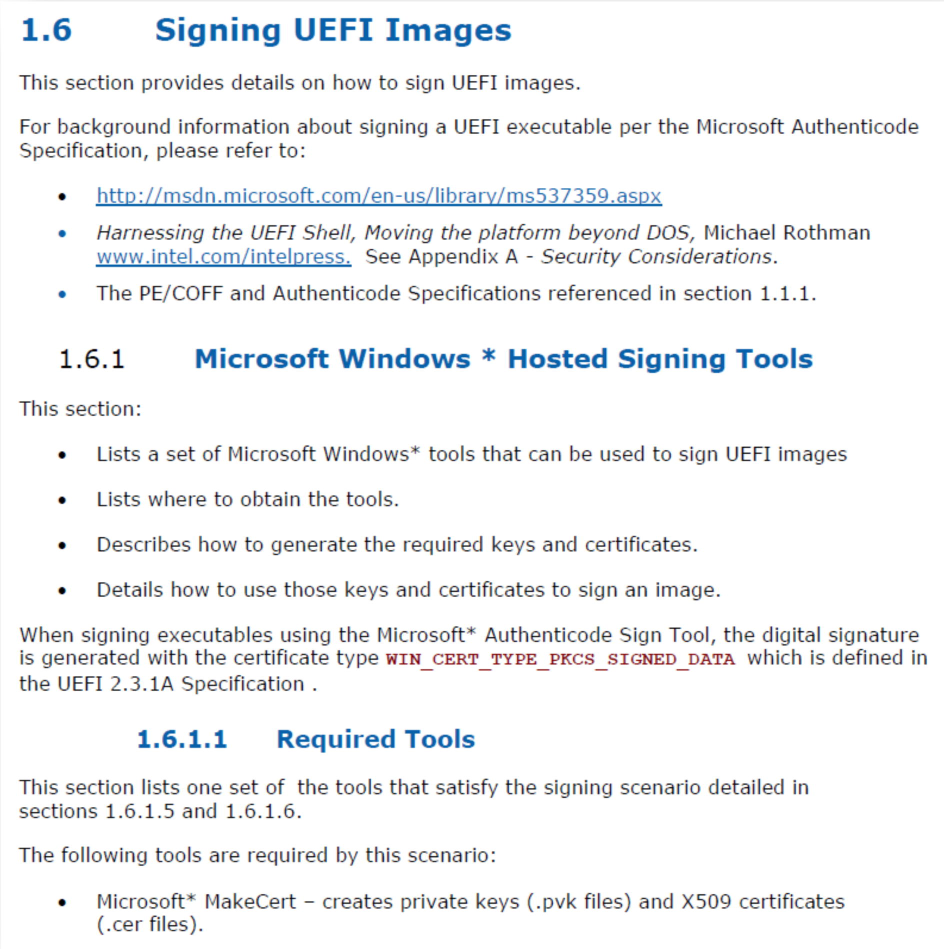
intel.com

- UEFI Driver Development Guides

Signing UEFI Applications and Drivers for UEFI Secure Boot

Recently added to tinocore.org

Describes UEFI Secure Boot & Driver Signing procedures using open source tools (EDK II)



1.6 Signing UEFI Images

This section provides details on how to sign UEFI images.

For background information about signing a UEFI executable per the Microsoft Authenticode Specification, please refer to:

- http://msdn.microsoft.com/en-us/library/ms537359.aspx
- *Harnessing the UEFI Shell, Moving the platform beyond DOS,* Michael Rothman www.intel.com/intelpress. See Appendix A - *Security Considerations*.
- The PE/COFF and Authenticode Specifications referenced in section 1.1.1.

1.6.1 Microsoft Windows * Hosted Signing Tools

This section:

- Lists a set of Microsoft Windows* tools that can be used to sign UEFI images
- Lists where to obtain the tools.
- Describes how to generate the required keys and certificates.
- Details how to use those keys and certificates to sign an image.

When signing executables using the Microsoft* Authenticode Sign Tool, the digital signature is generated with the certificate type **WIN_CERT_TYPE_PKCS_SIGNED_DATA** which is defined in the UEFI 2.3.1A Specification .

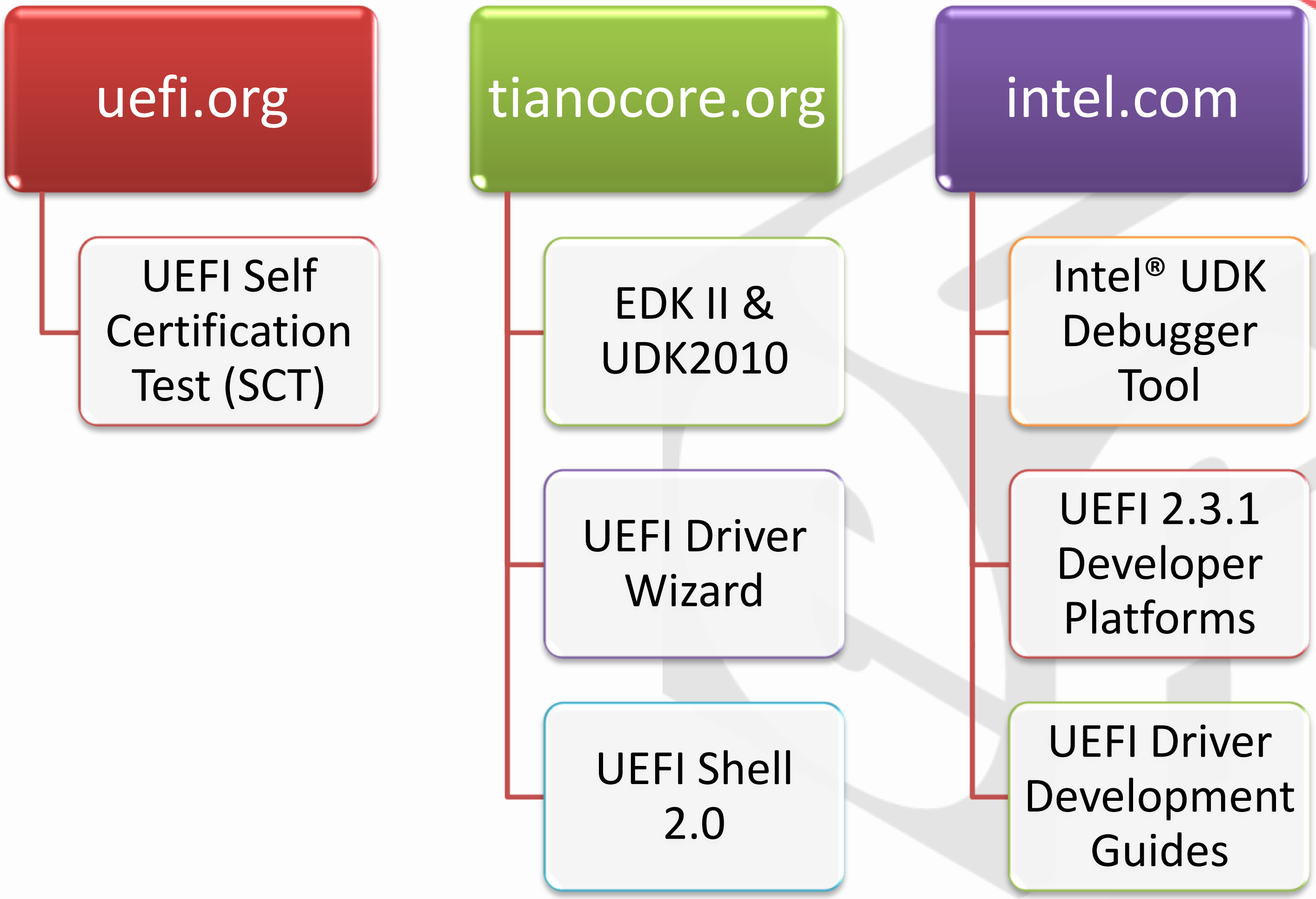
1.6.1.1 Required Tools

This section lists one set of the tools that satisfy the signing scenario detailed in sections 1.6.1.5 and 1.6.1.6.

The following tools are required by this scenario:

- Microsoft* MakeCert – creates private keys (.pvk files) and X509 certificates (.cer files).

Development Resources



uefi.org

- UEFI Self Certification Test (SCT)

tianocore.org

- EDK II & UDK2010
- UEFI Driver Wizard
- UEFI Shell 2.0

intel.com

- Intel® UDK Debugger Tool
- UEFI 2.3.1 Developer Platforms
- UEFI Driver Development Guides

UEFI 2.3.1 Developer Platforms

Q2 2012

Intel DQ57TM

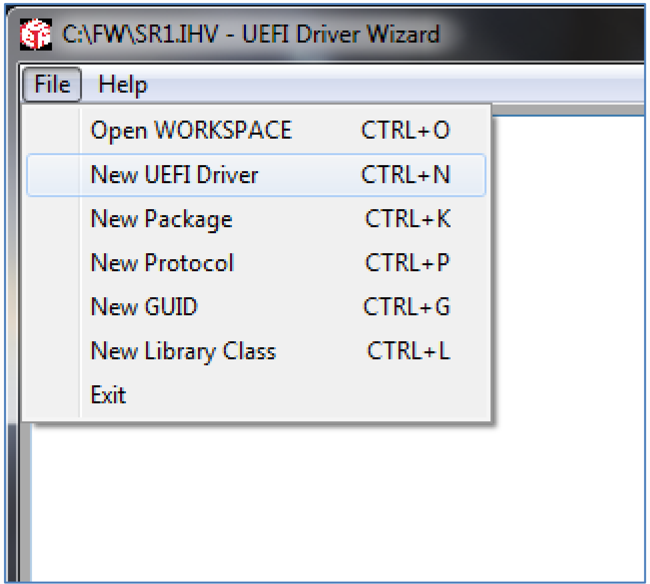
Intel DQ67SW

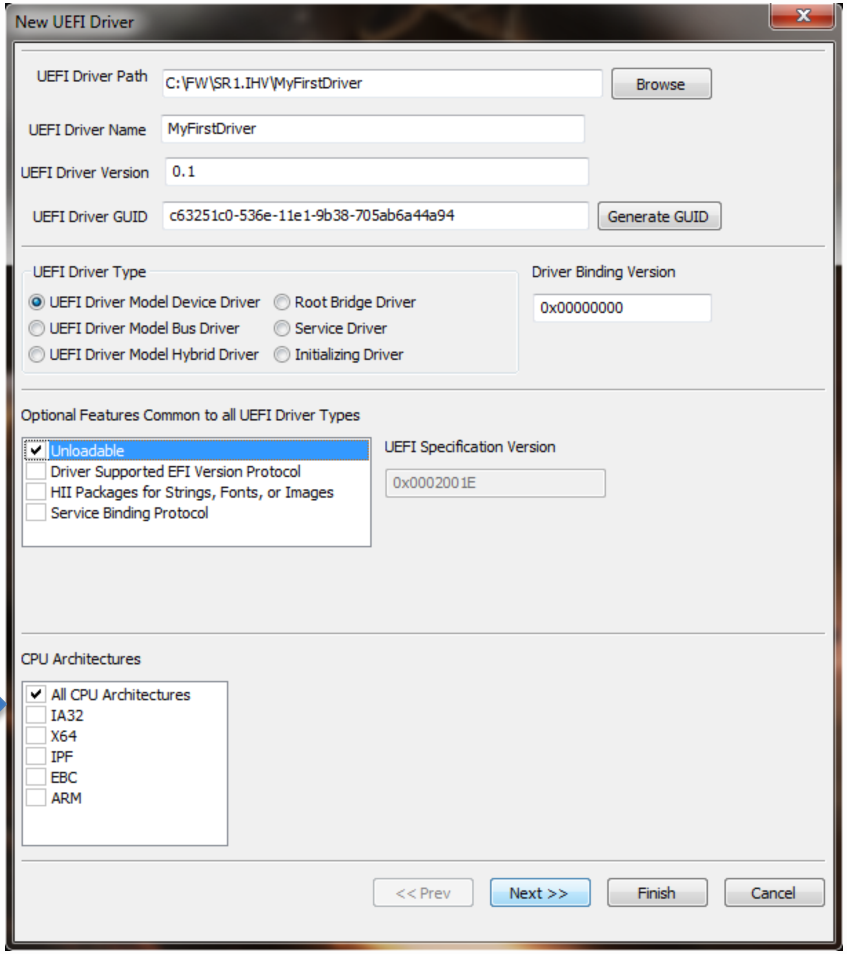
- Use to debug OS and add-in hardware against the latest UEFI functionality
  - UEFI 2.3.1
  - UDK2010.SR1+
  - UEFI Secure Boot
- Based on Intel production quality hardware with UEFI BIOS images
  - Release, debug & source-level debug versions

UEFI Driver Wizard

- Menu-based GUI designed to simplify UEFI Driver Development
  - Uses “IHV” subset of UDK2010
  - Wizard-based template generation
- Open source project contributed to tianocore.org by Intel SSG
  - Python interface, designed for extensibility
  - Intel encourages contribution by developers

UEFI Driver Wizard





UEFI Driver Path: C:\FW\SR1.IHV\MyFirstDriver

UEFI Driver Name: MyFirstDriver

UEFI Driver Version: 0.1

UEFI Driver GUID: c63251c0-536e-11e1-9b38-705ab6a44a94

UEFI Driver Type

- ● UEFI Driver Model Device Driver
- ○ UEFI Driver Model Bus Driver
- ○ UEFI Driver Model Hybrid Driver
- ○ Root Bridge Driver
- ○ Service Driver
- ○ Initializing Driver

Driver Binding Version: 0x00000000

Optional Features Common to all UEFI Driver Types

- ☑ Unloadable
- □ Driver Supported EFI Version Protocol
- □ HII Packages for Strings, Fonts, or Images
- □ Service Binding Protocol

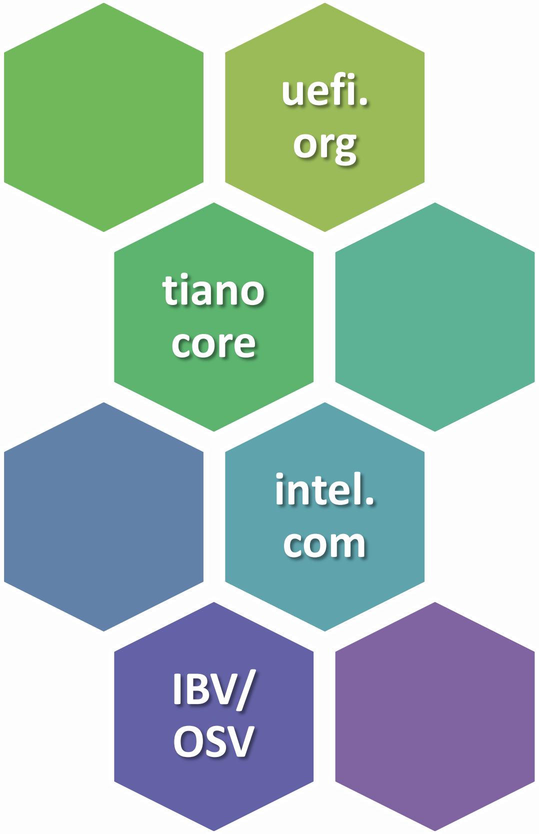
UEFI Specification Version: 0x0002001E

CPU Architectures

- ☑ All CPU Architectures
- □ IA32
- □ X64
- □ IPF
- □ EBC
- □ ARM

Demo

Problem: Finding Resources



uefi.org

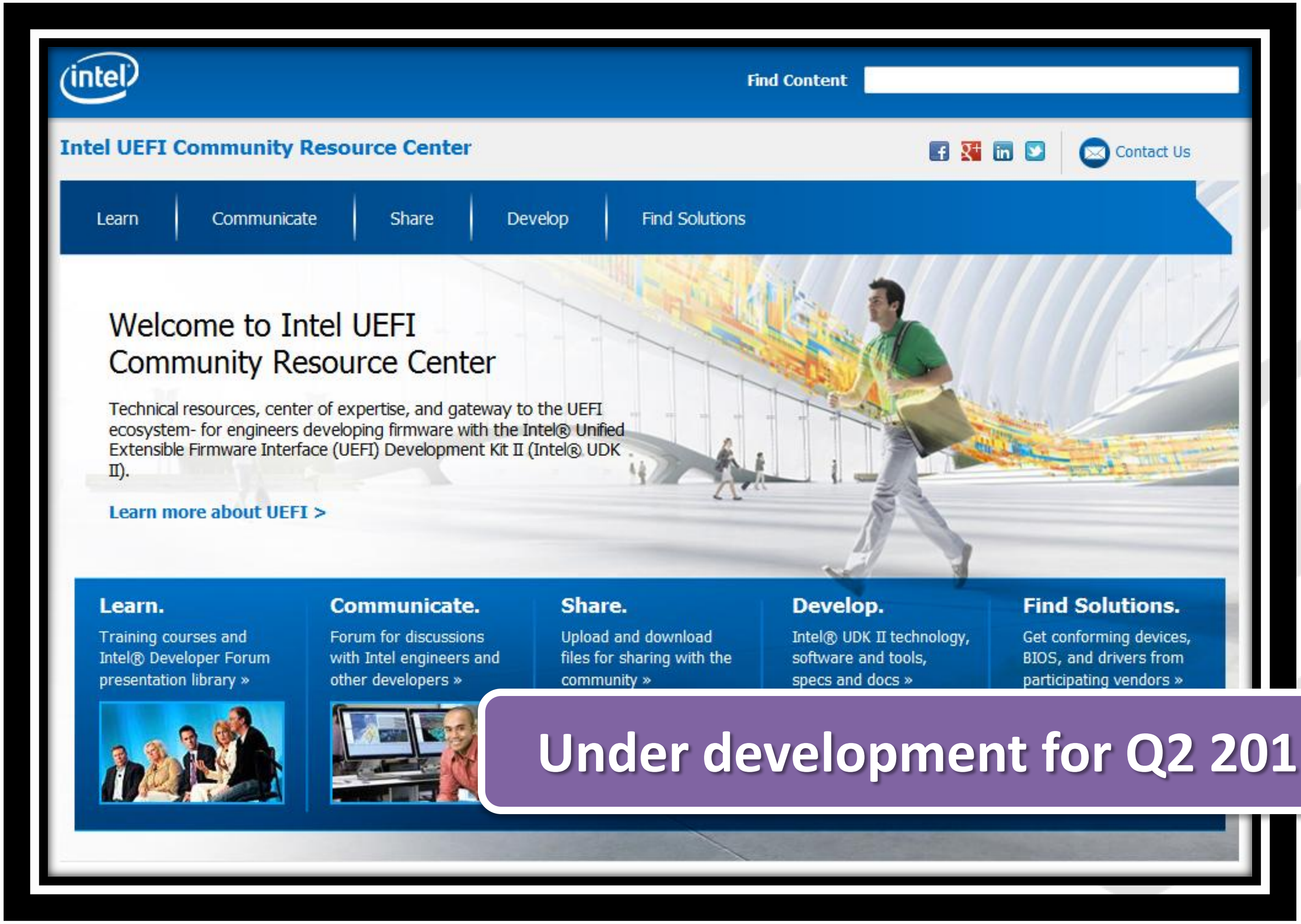
tianocore

intel.com

IBV/OSV

- UEFI resources are spread across multiple sites, making it harder for developers to find what they need
- BIOS vendors & software developers need a place to connect outside of the open source communities

The Intel UEFI Community Resource Center



intel

Find Content

Intel UEFI Community Resource Center

Contact Us

Learn | Communicate | Share | Develop | Find Solutions

Welcome to Intel UEFI Community Resource Center

Technical resources, center of expertise, and gateway to the UEFI ecosystem- for engineers developing firmware with the Intel® Unified Extensible Firmware Interface (UEFI) Development Kit II (Intel® UDK II).

Learn more about UEFI >

Learn.
Training courses and Intel® Developer Forum presentation library »

Communicate.
Forum for discussions with Intel engineers and other developers »

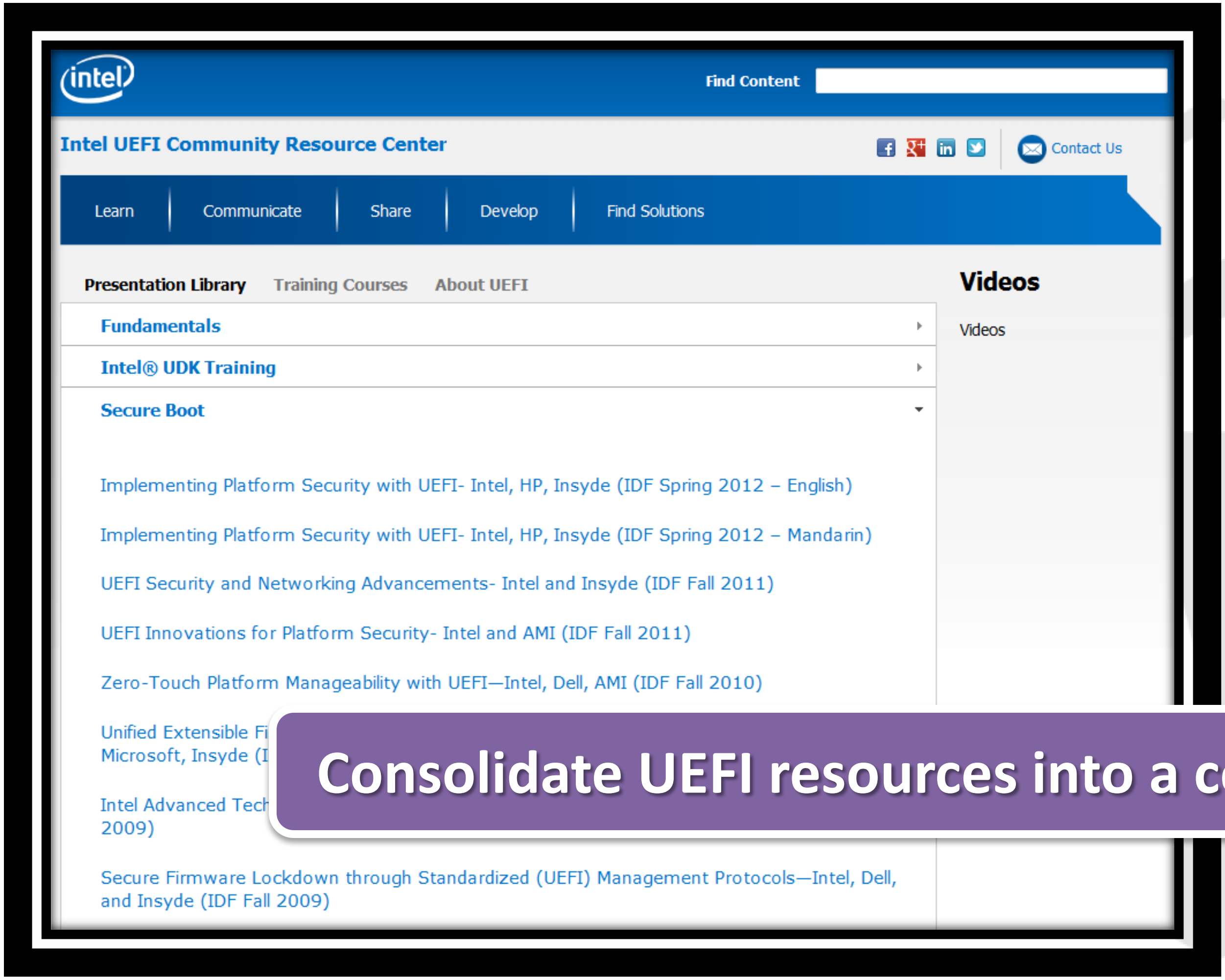
Share.
Upload and download files for sharing with the community »

Develop.
Intel® UDK II technology, software and tools, specs and docs »

Find Solutions.
Get conforming devices, BIOS, and drivers from participating vendors »

Under development for Q2 2012 launch

The Intel UEFI Community Resource Center



Intel UEFI Community Resource Center

Videos

Videos

Fundamentals

Intel® UDK Training

Secure Boot

Implementing Platform Security with UEFI- Intel, HP, Insyde (IDF Spring 2012 – English)

Implementing Platform Security with UEFI- Intel, HP, Insyde (IDF Spring 2012 – Mandarin)

UEFI Security and Networking Advancements- Intel and Insyde (IDF Fall 2011)

UEFI Innovations for Platform Security- Intel and AMI (IDF Fall 2011)

Zero-Touch Platform Manageability with UEFI—Intel, Dell, AMI (IDF Fall 2010)

Unified Extensible Fi
Microsoft, Insyde (I

Intel Advanced Tech
2009)

Secure Firmware Lockdown through Standardized (UEFI) Management Protocols—Intel, Dell, and Insyde (IDF Fall 2009)

Demo

Consolidate UEFI resources into a central community

Summary / Q&A

- UEFI Has a Robust Developer Community
  - Documentation Resources
  - Development Resources
  - Based on Open Source Projects & Member Company Contributions
- Intel adds the *Intel UEFI Community Resource Center* to aid UEFI development

Get More Information

- UEFI Forum Learning Center
  - http://www.uefi.org/learning_center/
- UEFI IHV Resources @ intel.com
  - http://intel.com/go/uefi-ihv
- Use the TianoCore *edk2-devel* mailing list for support from other UEFI developers
- Stay tuned for the launch of the Intel UEFI Community (Q2 2012)

Thanks for attending the
UEFI Spring Plugfest 2012

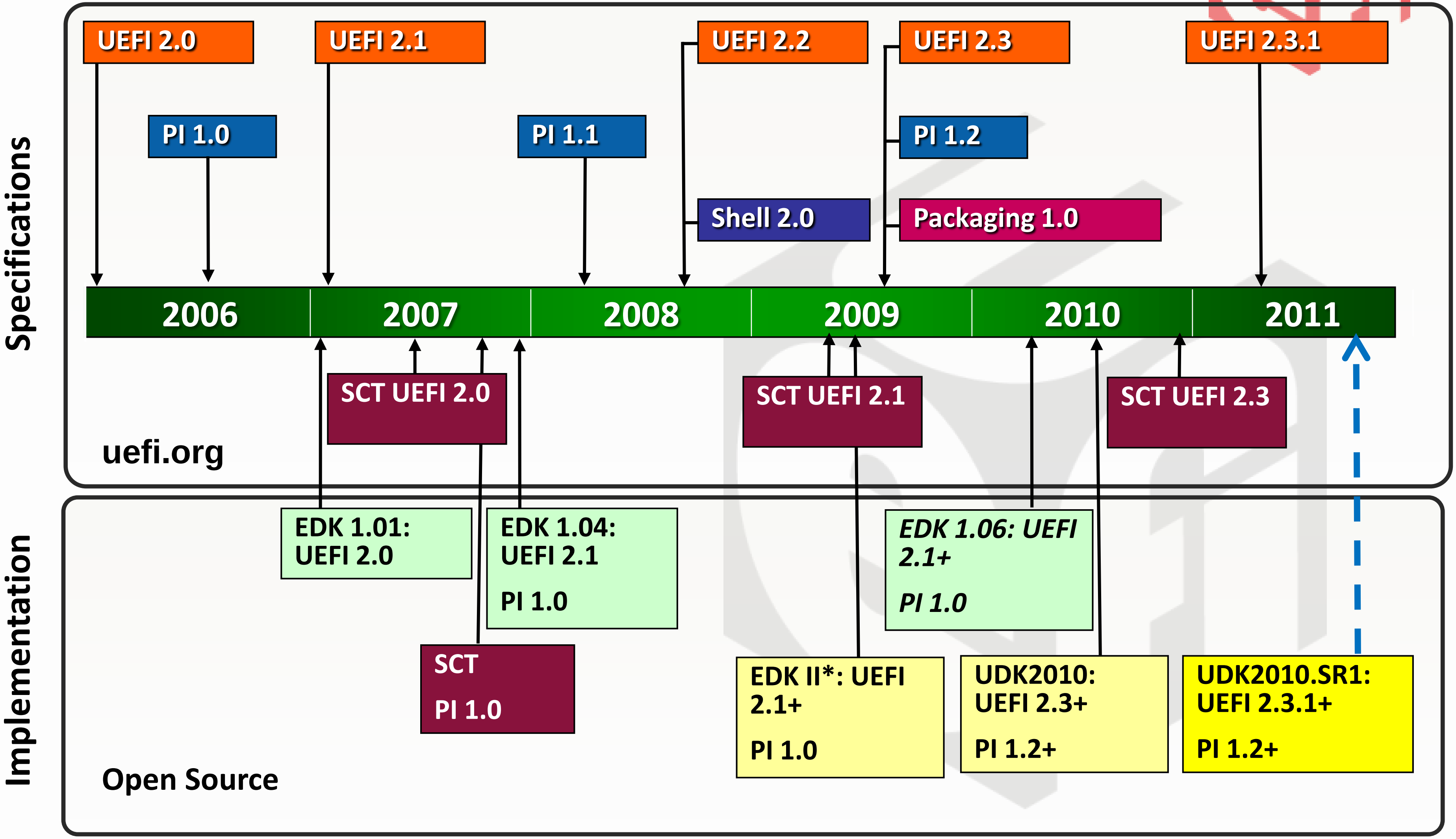
For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by

UEFI Development Community

Backup Slides

UEFI Specification Timeline



Specifications

UEFI 2.0

UEFI 2.1

UEFI 2.2

UEFI 2.3

UEFI 2.3.1

PI 1.0

PI 1.1

PI 1.2

Shell 2.0

Packaging 1.0

2006

2007

2008

2009

2010

2011

SCT UEFI 2.0

SCT UEFI 2.1

SCT UEFI 2.3

uefi.org

Implementation

EDK 1.01: UEFI 2.0

EDK 1.04: UEFI 2.1

PI 1.0

EDK 1.06: UEFI 2.1+

PI 1.0

SCT

PI 1.0

EDK II*: UEFI 2.1+

PI 1.0

UDK2010: UEFI 2.3+

PI 1.2+

UDK2010.SR1: UEFI 2.3.1+

PI 1.2+

Open Source

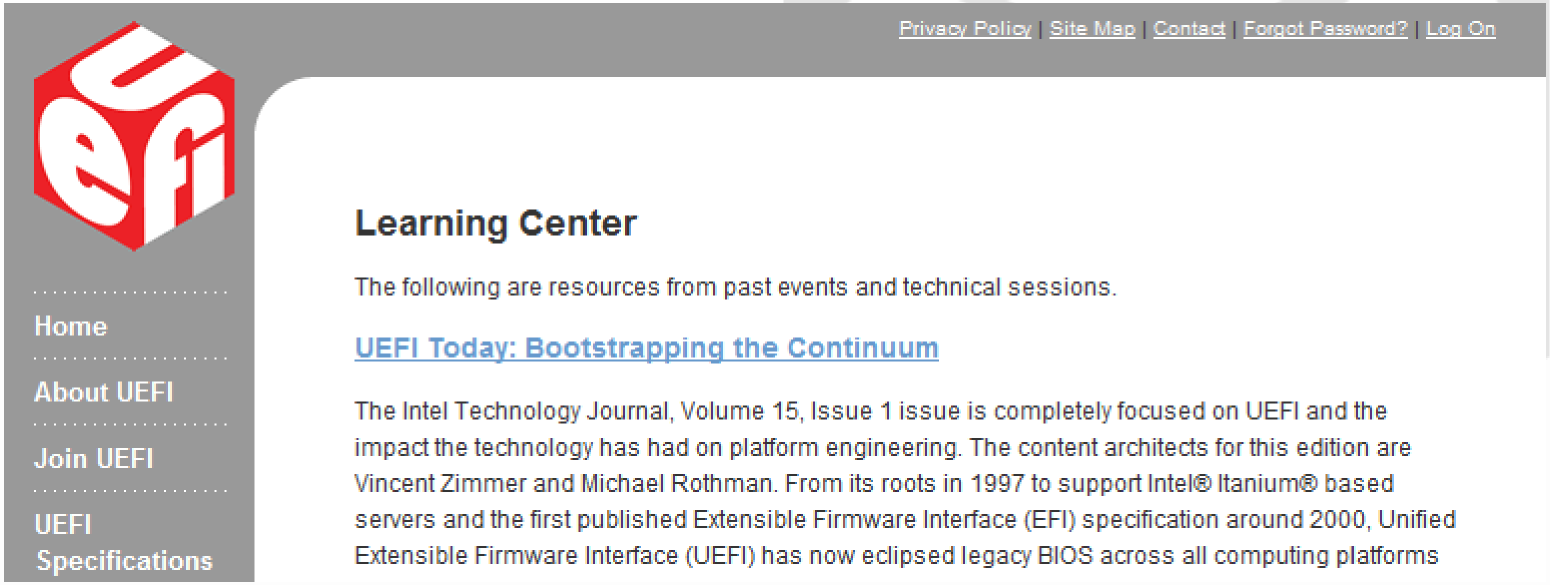
* EDK II is same code base as UDK2010

EDK II versus UDK2010

- EDK II is the open source “TianoCore” project
  - Available under BSD license at tianocore.org
- Intel SSG uses this project as the base for a common UEFI implementation within Intel
  - Intel® UEFI Development Kit 2010 (UDK2010)
  - UDK2010 is a stable snapshot of EDK II that has been validated against Intel silicon components
  - Most recent open-source release is UDK2010.SR1
- *EDK II rev 12898* is the base for UDK2010.SR1

UEFI Learning Center

- http://www.uefi.org/learning_center/
  - Related journals & whitepapers
  - Presentations from UEFI Plugfests



Privacy Policy | Site Map | Contact | Forgot Password? | Log On
Home
About UEFI
Join UEFI
UEFI Specifications

Learning Center

The following are resources from past events and technical sessions.

UEFI Today: Bootstrapping the Continuum

The Intel Technology Journal, Volume 15, Issue 1 issue is completely focused on UEFI and the impact the technology has had on platform engineering. The content architects for this edition are Vincent Zimmer and Michael Rothman. From its roots in 1997 to support Intel® Itanium® based servers and the first published Extensible Firmware Interface (EFI) specification around 2000, Unified Extensible Firmware Interface (UEFI) has now eclipsed legacy BIOS across all computing platforms

UEFI Driver Writer's Guide

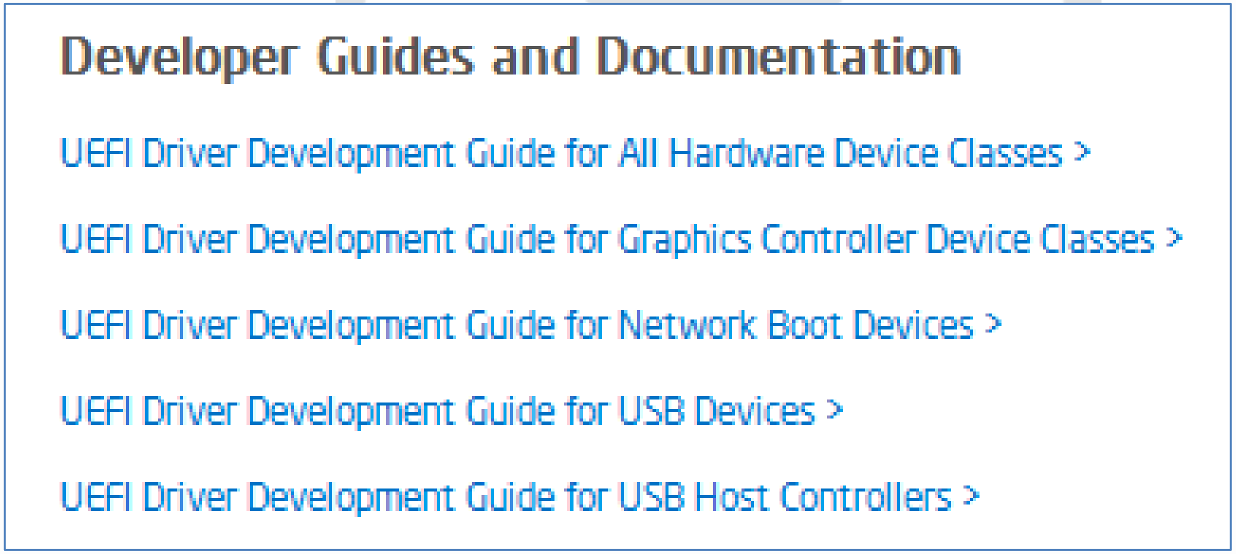
- Updated by Intel in Feb 2012
- Expanded to cover UEFI 2.3+ topics
- Designed as a developer reference
  - Organized & indexed by driver function
  - Not a "cover to cover read"
- http://intel.com/go/uefi-ihv

A comprehensive resource for UEFI Driver Developers …

Driver Development Guides

- Published by Intel in Nov 2011
- Supplements for specific driver classes
- http://intel.com/go/uefi-ihv

Short resources to help developers get started with UEFI drivers …



Developer Guides and Documentation

UEFI Driver Development Guide for All Hardware Device Classes >

UEFI Driver Development Guide for Graphics Controller Device Classes >

UEFI Driver Development Guide for Network Boot Devices >

UEFI Driver Development Guide for USB Devices >

UEFI Driver Development Guide for USB Host Controllers >

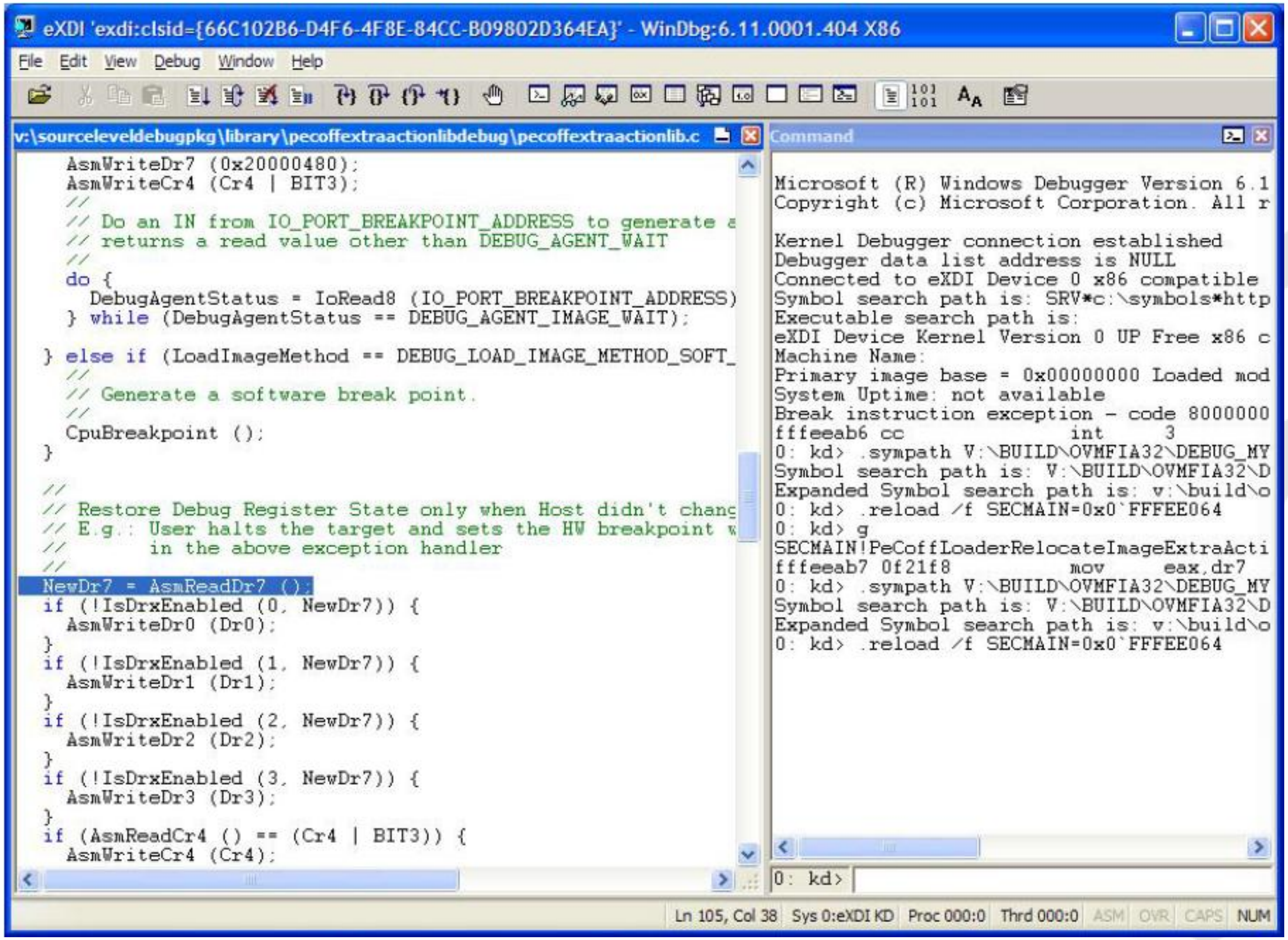
Open Source Resources

- Community for core UEFI components in open-source - http://tianocore.org
  - Develop firmware, drivers & applications
- Main TianoCore Projects
  - EDK Development Kit (EDK II)
  - UEFI Development Kit (UDK2010)
  - UEFI Shell

Intel® UDK Debugger Tool

- Software debugger for UEFI & EDK II
  - Connect via COM or USB Debug Port
  - Supports Microsoft Windows (WinDBG) and Linux (gdb) OS environments
  - Target side agent available in the EDK II `SourceLevelDebugPkg` component
- http://intel.com/go/uefi-ihv

Intel® UDK Debugger Tool


eXDI 'exdi:clsid={66C102B6-D4F6-4F8E-84CC-B09802D364EA}' - WinDbg:6.11.0001.404 X86
File Edit View Debug Window Help
v:\sourceleveldebugpkg\library\pecoffextraactionlibdebug\pecoffextraactionlib.c
Command
Ln 105, Col 38
Sys 0:eXDI KD
Proc 000:0
Thrd 000:0
ASM
OVR
CAPS
NUM

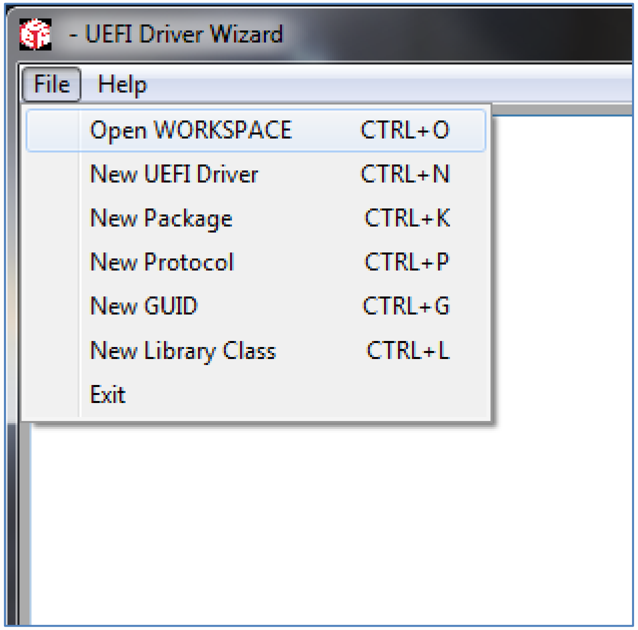
```
    AsmWriteDr7 (0x20000480);
    AsmWriteCr4 (Cr4 | BIT3);
    //
    // Do an IN from IO_PORT_BREAKPOINT_ADDRESS to generate a
    // returns a read value other than DEBUG_AGENT_WAIT
    //
    do {
      DebugAgentStatus = IoRead8 (IO_PORT_BREAKPOINT_ADDRESS)
    } while (DebugAgentStatus == DEBUG_AGENT_IMAGE_WAIT);

  } else if (LoadImageMethod == DEBUG_LOAD_IMAGE_METHOD_SOFT_
    //
    // Generate a software break point.
    //
    CpuBreakpoint ();
  }

  //
  // Restore Debug Register State only when Host didn't chang
  // E.g.: User halts the target and sets the HW breakpoint w
  //       in the above exception handler
  //
  NewDr7 = AsmReadDr7 ();
  if (!IsDrxEnabled (0, NewDr7)) {
    AsmWriteDr0 (Dr0);
  }
  if (!IsDrxEnabled (1, NewDr7)) {
    AsmWriteDr1 (Dr1);
  }
  if (!IsDrxEnabled (2, NewDr7)) {
    AsmWriteDr2 (Dr2);
  }
  if (!IsDrxEnabled (3, NewDr7)) {
    AsmWriteDr3 (Dr3);
  }
  if (AsmReadCr4 () == (Cr4 | BIT3)) {
    AsmWriteCr4 (Cr4);
```

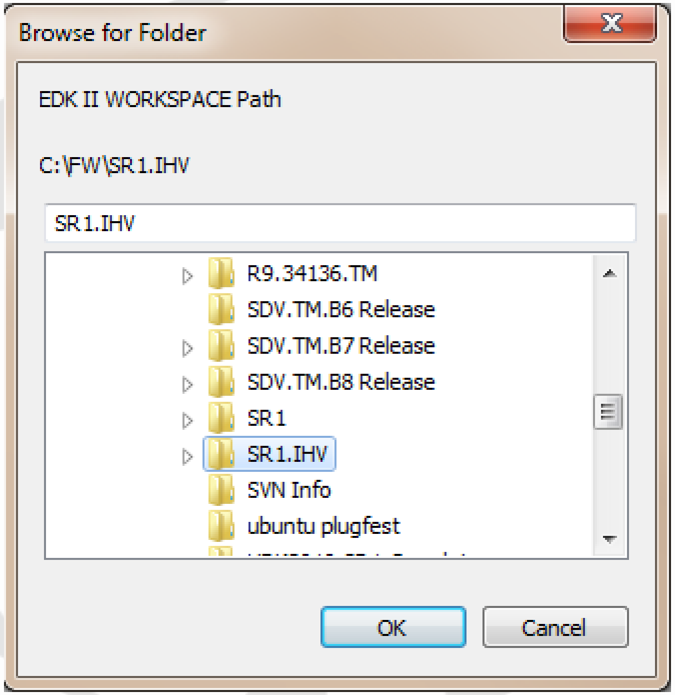
```
Microsoft (R) Windows Debugger Version 6.1
Copyright (c) Microsoft Corporation. All r

Kernel Debugger connection established
Debugger data list address is NULL
Connected to eXDI Device 0 x86 compatible
Symbol search path is: SRV*c:\symbols*http
Executable search path is:
eXDI Device Kernel Version 0 UP Free x86 c
Machine Name:
Primary image base = 0x00000000 Loaded mod
System Uptime: not available
Break instruction exception - code 8000000
fffeeab6 cc              int     3
0: kd> .sympath V:\BUILD\OVMFIA32\DEBUG_MY
Symbol search path is: V:\BUILD\OVMFIA32\D
Expanded Symbol search path is: v:\build\o
0: kd> .reload /f SECMAIN=0x0`FFFEE064
0: kd> g
SECMAIN!PeCoffLoaderRelocateImageExtraActi
fffeeab7 0f21f8          mov     eax,dr7
0: kd> .sympath V:\BUILD\OVMFIA32\DEBUG_MY
Symbol search path is: V:\BUILD\OVMFIA32\D
Expanded Symbol search path is: v:\build\o
0: kd> .reload /f SECMAIN=0x0`FFFEE064
```

```
0: kd>
```

WinDBG

Screenshots from the UEFI Driver Wizard



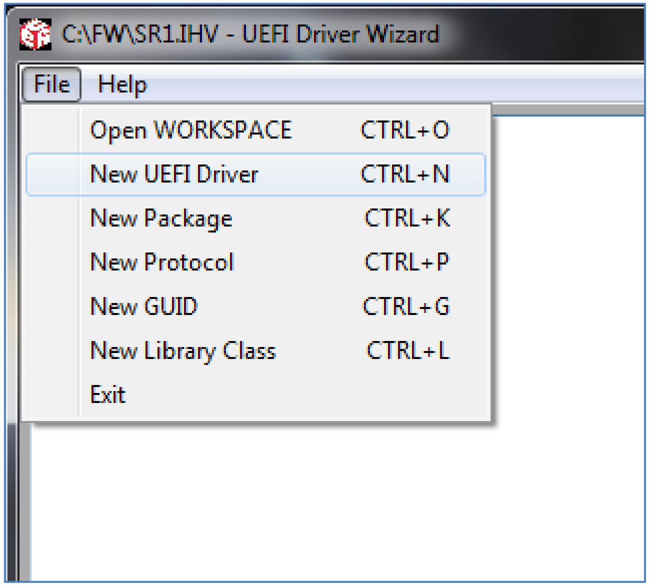


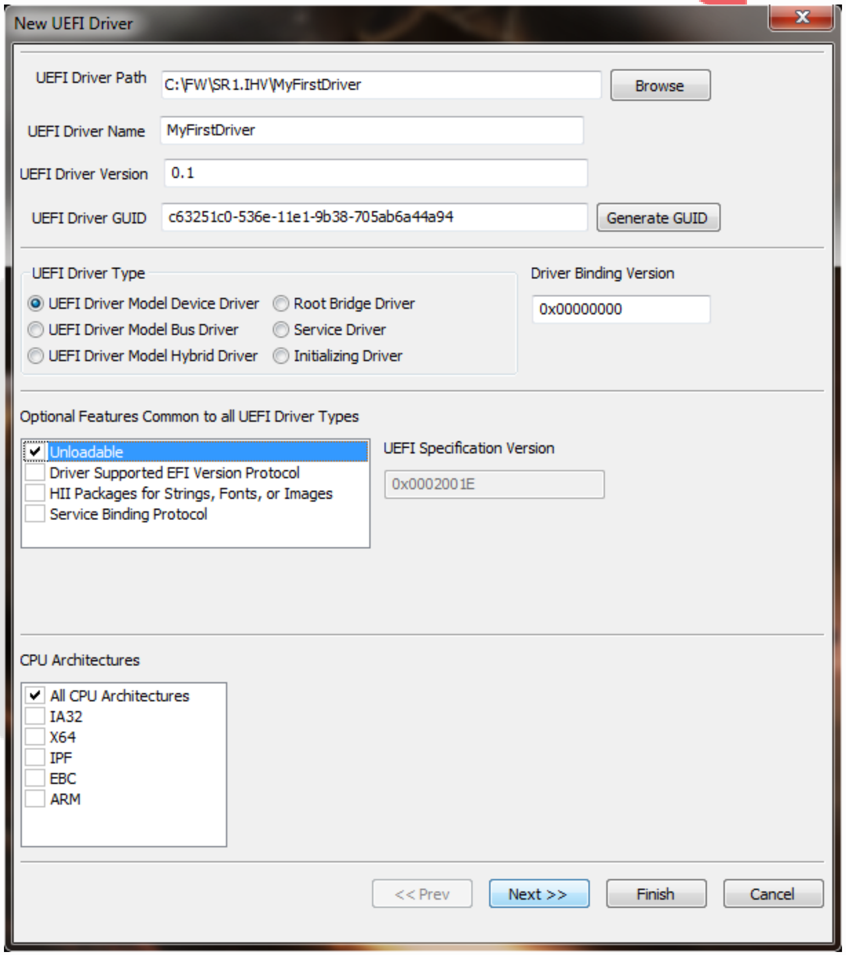
EDK II WORKSPACE Path

C:\FW\SR1.IHV

SR1.IHV

Screenshots from the UEFI Driver Wizard




New UEFI Driver

UEFI Driver Path	C:\FW\SR1.IHV\MyFirstDriver
UEFI Driver Name	MyFirstDriver
UEFI Driver Version	0.1
UEFI Driver GUID	c63251c0-536e-11e1-9b38-705ab6a44a94

UEFI Driver Type

- ● UEFI Driver Model Device Driver
- ○ UEFI Driver Model Bus Driver
- ○ UEFI Driver Model Hybrid Driver
- ○ Root Bridge Driver
- ○ Service Driver
- ○ Initializing Driver

Driver Binding Version: 0x00000000

Optional Features Common to all UEFI Driver Types

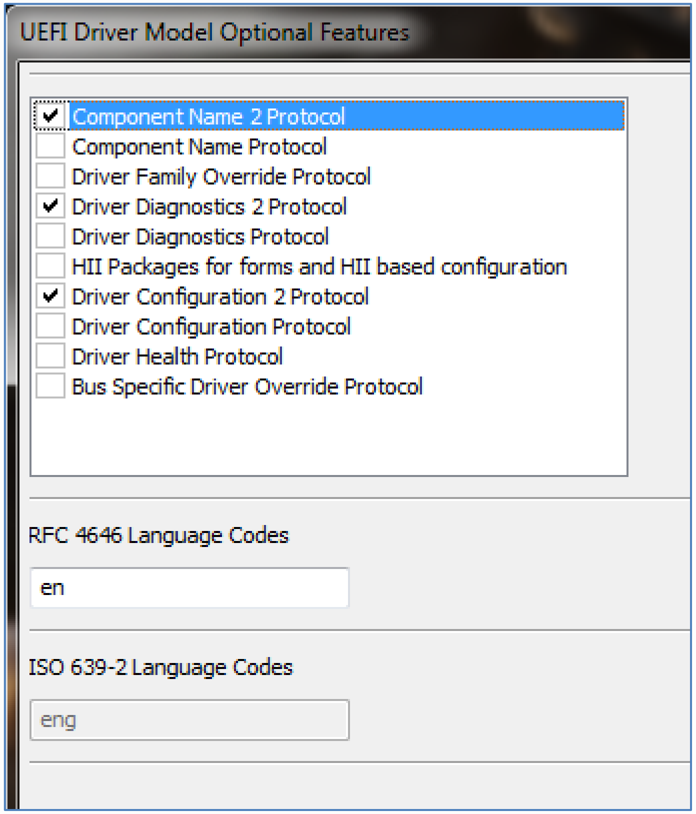
- ☑ Unloadable
- □ Driver Supported EFI Version Protocol
- □ HII Packages for Strings, Fonts, or Images
- □ Service Binding Protocol

UEFI Specification Version: 0x0002001E

CPU Architectures

- ☑ All CPU Architectures
- □ IA32
- □ X64
- □ IPF
- □ EBC
- □ ARM

Screenshots from the UEFI Driver Wizard

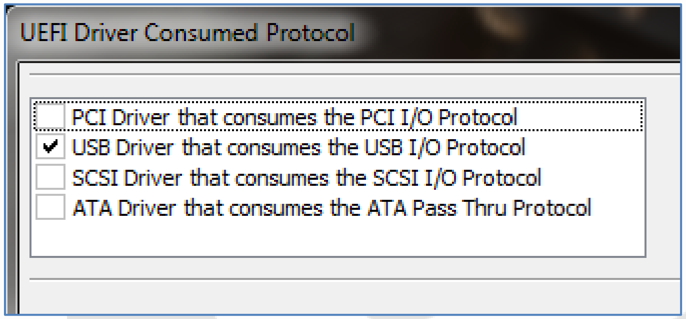


UEFI Driver Model Optional Features

- ☑ Component Name 2 Protocol
- □ Component Name Protocol
- □ Driver Family Override Protocol
- ☑ Driver Diagnostics 2 Protocol
- □ Driver Diagnostics Protocol
- □ HII Packages for forms and HII based configuration
- ☑ Driver Configuration 2 Protocol
- □ Driver Configuration Protocol
- □ Driver Health Protocol
- □ Bus Specific Driver Override Protocol

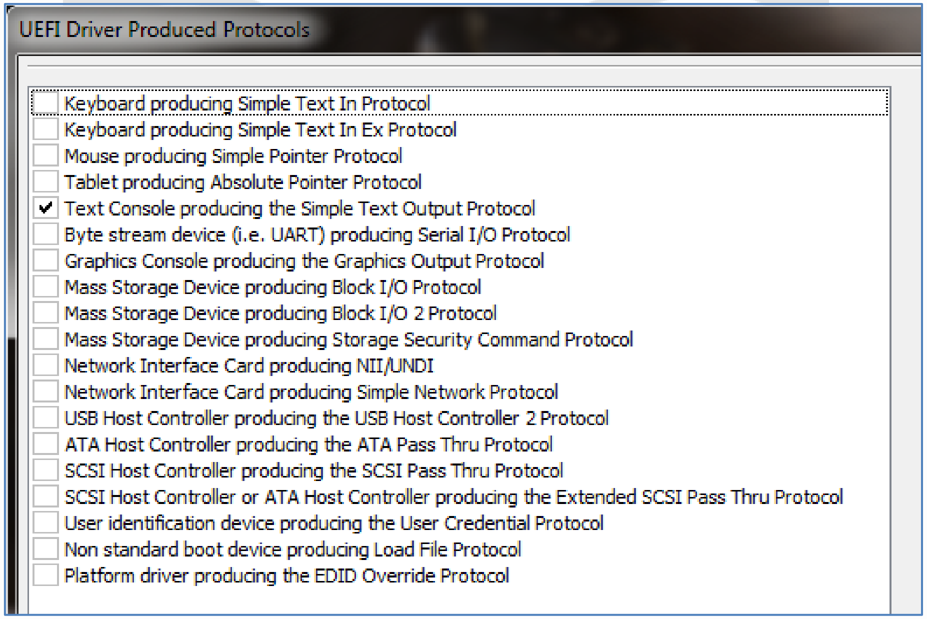
RFC 4646 Language Codes: en

ISO 639-2 Language Codes: eng



UEFI Driver Consumed Protocol

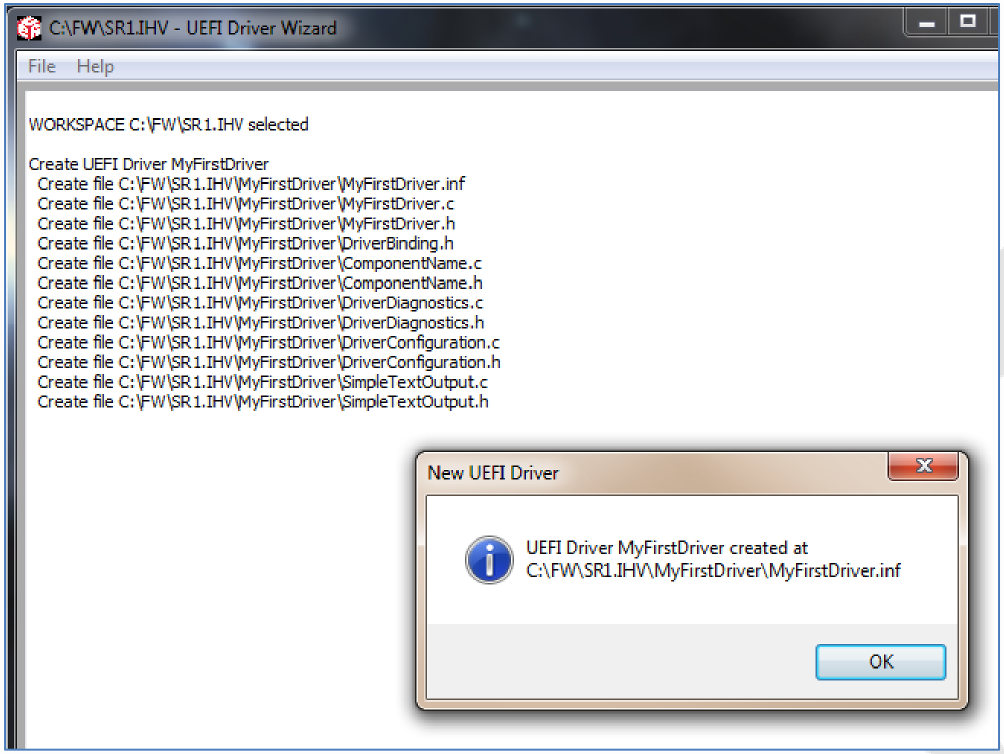
- □ PCI Driver that consumes the PCI I/O Protocol
- ☑ USB Driver that consumes the USB I/O Protocol
- □ SCSI Driver that consumes the SCSI I/O Protocol
- □ ATA Driver that consumes the ATA Pass Thru Protocol



UEFI Driver Produced Protocols

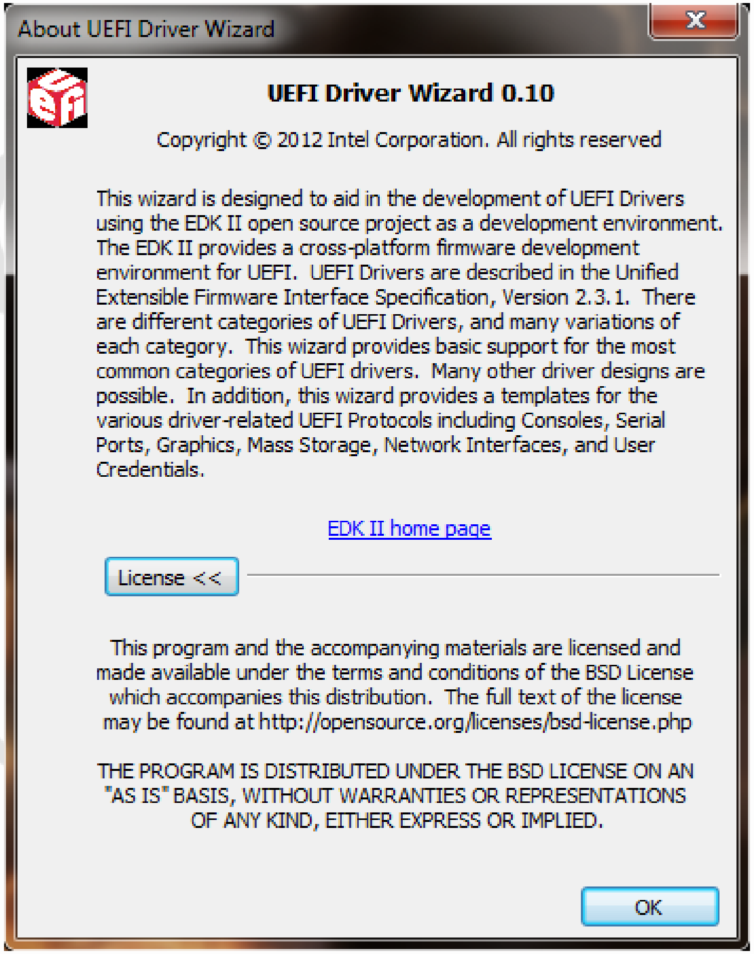
- □ Keyboard producing Simple Text In Protocol
- □ Keyboard producing Simple Text In Ex Protocol
- □ Mouse producing Simple Pointer Protocol
- □ Tablet producing Absolute Pointer Protocol
- ☑ Text Console producing the Simple Text Output Protocol
- □ Byte stream device (i.e. UART) producing Serial I/O Protocol
- □ Graphics Console producing the Graphics Output Protocol
- □ Mass Storage Device producing Block I/O Protocol
- □ Mass Storage Device producing Block I/O 2 Protocol
- □ Mass Storage Device producing Storage Security Command Protocol
- □ Network Interface Card producing NII/UNDI
- □ Network Interface Card producing Simple Network Protocol
- □ USB Host Controller producing the USB Host Controller 2 Protocol
- □ ATA Host Controller producing the ATA Pass Thru Protocol
- □ SCSI Host Controller producing the SCSI Pass Thru Protocol
- □ SCSI Host Controller or ATA Host Controller producing the Extended SCSI Pass Thru Protocol
- □ User identification device producing the User Credential Protocol
- □ Non standard boot device producing Load File Protocol
- □ Platform driver producing the EDID Override Protocol

Screenshots from the UEFI Driver Wizard



```
WORKSPACE C:\FW\SR1.IHV selected

Create UEFI Driver MyFirstDriver
  Create file C:\FW\SR1.IHV\MyFirstDriver\MyFirstDriver.inf
  Create file C:\FW\SR1.IHV\MyFirstDriver\MyFirstDriver.c
  Create file C:\FW\SR1.IHV\MyFirstDriver\MyFirstDriver.h
  Create file C:\FW\SR1.IHV\MyFirstDriver\DriverBinding.h
  Create file C:\FW\SR1.IHV\MyFirstDriver\ComponentName.c
  Create file C:\FW\SR1.IHV\MyFirstDriver\ComponentName.h
  Create file C:\FW\SR1.IHV\MyFirstDriver\DriverDiagnostics.c
  Create file C:\FW\SR1.IHV\MyFirstDriver\DriverDiagnostics.h
  Create file C:\FW\SR1.IHV\MyFirstDriver\DriverConfiguration.c
  Create file C:\FW\SR1.IHV\MyFirstDriver\DriverConfiguration.h
  Create file C:\FW\SR1.IHV\MyFirstDriver\SimpleTextOutput.c
  Create file C:\FW\SR1.IHV\MyFirstDriver\SimpleTextOutput.h
```

UEFI Driver MyFirstDriver created at
C:\FW\SR1.IHV\MyFirstDriver\MyFirstDriver.inf



UEFI Driver Wizard 0.10

Copyright © 2012 Intel Corporation. All rights reserved

This wizard is designed to aid in the development of UEFI Drivers using the EDK II open source project as a development environment. The EDK II provides a cross-platform firmware development environment for UEFI. UEFI Drivers are described in the Unified Extensible Firmware Interface Specification, Version 2.3.1. There are different categories of UEFI Drivers, and many variations of each category. This wizard provides basic support for the most common categories of UEFI drivers. Many other driver designs are possible. In addition, this wizard provides a templates for the various driver-related UEFI Protocols including Consoles, Serial Ports, Graphics, Mass Storage, Network Interfaces, and User Credentials.

This program and the accompanying materials are licensed and made available under the terms and conditions of the BSD License which accompanies this distribution. The full text of the license may be found at http://opensource.org/licenses/bsd-license.php

THE PROGRAM IS DISTRIBUTED UNDER THE BSD LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR REPRESENTATIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED.

UEFI Development Resources

www.uefi.org