IDF2011
INTEL DEVELOPER FORUM

Beyond DOS: UEFI Modern Pre-boot Application Development Environment

Tim Lewis, Chief BIOS Architect, Phoenix Technologies Ltd
Vincent Zimmer, Principal Engineer, Intel Corporation

EFIS003

Sponsors of Tomorrow.™ intel

Please Fill out the Online Session Evaluation Form

Be entered to win fabulous prizes every day!

Winners will be announced at 6pm (Day 1/2) and 3:30pm (Day 3)

You will receive an email prior to the end of this session.

IDF2011
INTEL DEVELOPER FORUM

Agenda

- Pre-Boot Application Opportunities
- Basic UEFI Application Programming
- UEFI Applications for Manufacturing
- UEFI Applications for Graphics

The PDF for this Session presentation is available from our Technical Session Catalog at the end of the day at: intel.com/go/idfsessions

URL is on top of Session Agenda Pages in Pocket Guide

Agenda

- Pre-Boot Application Opportunities
- Basic UEFI Application Programming
- UEFI Applications for Manufacturing
- UEFI Applications for Graphics

What Is Pre-OS Apps Opportunity?

- **The pre-OS environment offers unique opportunities for application development**
  - **hardware configuration, first user visibility**

- **UEFI offers a stable, standard, secure environment**
  - **Anti-virus**
  - **Recovery**
  - **Diagnostics**
  - **Disk imaging**

How Can We Make It Easier?

What is the UEFI Shell?

The UEFI Shell is a standardized programming environment and command-line interface that sits on top of UEFI 2.1+ firmware.

Scalable → Embeddable. Use only as much as you need, with profiles and support levels.

Scriptable → Familiar scripting interface. Automates repetitive tasks.

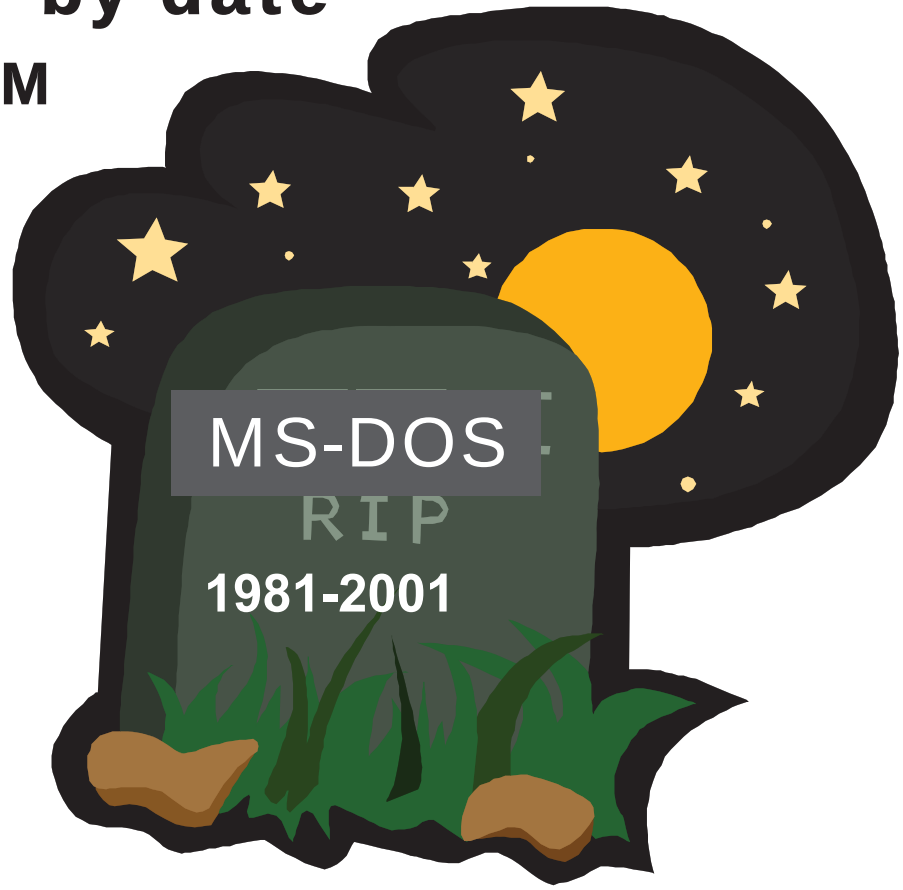
Standardized → Works on all UEFI Shell 2.0-compliant platforms. APIs and cmd. Line parameters.

The UEFI Shell 2.0

- Currently Spec. At 2.0
- Versions available now from tianocore.org or your BIOS vendor
- Described in the book: *Harnessing The UEFI Shell* (Intel Press)

Why Not DOS*?

- DOS – Well beyond its sell-by date
  - Barely supports >1MiB of RAM
  - Many devices not supported
  - Relies on the CSM in UEFI
    - Which is going away
  - Not supported & no updates
  - No up-to-date networking
    - How support IPv6?



The UEFI Shell is Scalable

- **Standard Shell Support Levels describe core capabilities**
- **Detect using environment variable**

Support Level	Description	Size Delta
0	Shell API, No Scripting	+82 KB
1	Scripting	+16 KB
2	File I/O	+43 KB
3	Console Input/Output	+12 KB

UEFI Shell Profiles

- **Standard Shell Profiles describe mix-and-match add-on capabilities**
  - **Detectable using environment variable**
  - **No dependencies between profiles**
  - **You can create your own profiles**

Profile Name	Description	Size Delta
Debug1	Debug commands.	(+144KB)
Network1	Network management commands.	(+24KB)
Driver1	Driver management commands.	(+68KB
Install1	Driver/application installation aid commands.	(+12KB)

The UEFI Shell is Scriptable

- Scripts use the file extension .NSH
  - startup.nsh automatically run when shell starts
- Similar to Windows* & DOS Command-Prompt batch files
- Extensions for:
  - Diskless Operation
    - i.e., Output Redirect to Environment Variable
  - Parsing files

The UEFI Shell is Standardized

- **UEFI Advantages = UEFI Shell Advantages**
  - **Flat memory model**
  - **Robust, extensible architecture**
  - **File system, Network, Keyboard, Mouse**
- **UEFI Works = UEFI Shell Works**
  - **No additional requirements to run**
- **Write Once/Run Anywhere For Pre-OS**
  - **With standard API/commands = applications from different ISVs work on any platform (Ex: UEFI SCT)**

Example #1: Script Detects Shell Capabilities

```
# check that Shell supports level 3 commands.
if %shellsupport% ult 3
  echo Must support UEFI Shell, Level 3
endif

# check that Shell supports Debug1 profile.
if profile(Debug1)
  echo UEFI Shell supports Debug1 profile
endif
```

Example #2: Script Parses Standardized Output

```
# parse out file data from ‘ls’
ls –sfo | parse FileInfo 2
ls –sfo | parse FileInfo 4
```

Option –sfo Standard Format Option

Agenda

- What is the UEFI Shell?
- Basic UEFI Application Programming
- UEFI Applications for Manufacturing
- UEFI Applications for Graphics

Example #3: Hello, World

HelloWorld.c

```
#include <stdio.h>

int
main (
  IN int argc,
  IN char *argv[]
  )
{
  printf("Hello World\n");
  return 0;
}
```

Shell Prompt

```
2.0 F12:\> helloworld
Hello World
2.0 F12:\> cls 3
```

```
2.0 F12:\> helloworld
Hello World
```

Porting MS-DOS* to UEFI Shell 2.0

- Simplest way to create a new UEFI application is to port over an existing UEFI application!
- Example: Phoenix ACPI Disassembler (AD.EXE)
  - Disassemble ACPI tables from files or system memory
  - 15K lines of C (no assembly)
  - Works under MS-DOS (using a DOS Extender DOS4GW) and Windows command-line
  - Compiled under Watcom C/C++ and Visual C++ 6.0
  - Uses standard C library functions supported under both compilers

What Did It Take?

Porting AD to UEFI Shell

- **Goals For Move To The UEFI Shell 2.0**
  - **Build 32-bit or 64-bit**
  - **Remove dependencies on DOS4GW**
  - **Update ACPI Table discovery mechanism to use UEFI**
  - **No missing features**
- **Steps for Porting**
  1. **Create new INF file (AD.INF) for build description**
     - **Include paths set to C library build path**
  2. **Remove DOS4GW-related code**
  3. **Update ACPI Table discovery to use EFI System Configuration Table**
  4. **Rebuild!**

Issues Found During Porting

- **64-bit portability issues**
  - `sizeof(int) != sizeof(void*)`
- **Better error checking by the compiler**
  - Dead code, uninitialized variables
  - Fixed the bugs I didn't know I had
- **Used ISO C/C++ Library Functions**
  - Microsoft* C/C++ Library uses ISO C++ Names
  - UDK C Library uses ANSI C95 Names
  - Changed to Compliant Names
    - i.e from _open to open

Issues Found During Porting (2)

- **Used Microsoft*-specific Functions/Headers**
  - `_splitpath, _makepath`
  - `io.h/process.h/malloc.h`
  - **Created Equivalent Functions**

1 Day Effort To Move To UEFI Shell

Agenda

- What is the UEFI Shell?
- Basic UEFI Application Programming
- UEFI Applications for Manufacturing
- UEFI Applications for Graphics

UEFI Shell and Manufacturing

- **Many manufacturing tasks are simplified by:**
  - ***Getting*** **the latest tools / resources from the network**
  - ***Using*** **the latest tools / resources from the UEFI Shell**
  - **Let's combine the best of both worlds by…**
- **Launching the UEFI Shell remotely**
  - **From Windows* 2008 Server or other configured OS**
  - **Without impact to standard remote booting**
  - **With only minimal changes on server side**
  - **Without mounting a remote file system**

Remote Boot Overview

Intel® Desktop Board
DQ57TML

phoenix technologies

IDF2011
INTEL DEVELOPER FORUM

Power On

Intel® Desktop Board
DQ57TML

Issue DHCP Request

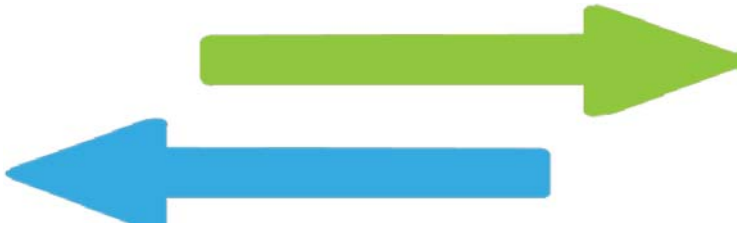
Intel® Desktop Board
DQ57TML

DHCP Reply

Intel® Desktop Board
DQ57TML

Boot Server Download Request

Intel® Desktop Board
DQ57TML



Boot Loader

Intel® Desktop Board
DQ57TML

Verify and Boot

Intel® Desktop Board
DQ57TML

Booting The Shell Remotely

Intel® Desktop Board
DQ57TML

phoenix technologies

IDF2011
INTEL DEVELOPER FORUM

Load File Request

Intel® Desktop Board
DQ57TML

Server Sends Replacement Boot Loader

Intel® Desktop Board
DQ57TML

phoenix technologies

IDF2011
INTEL DEVELOPER FORUM

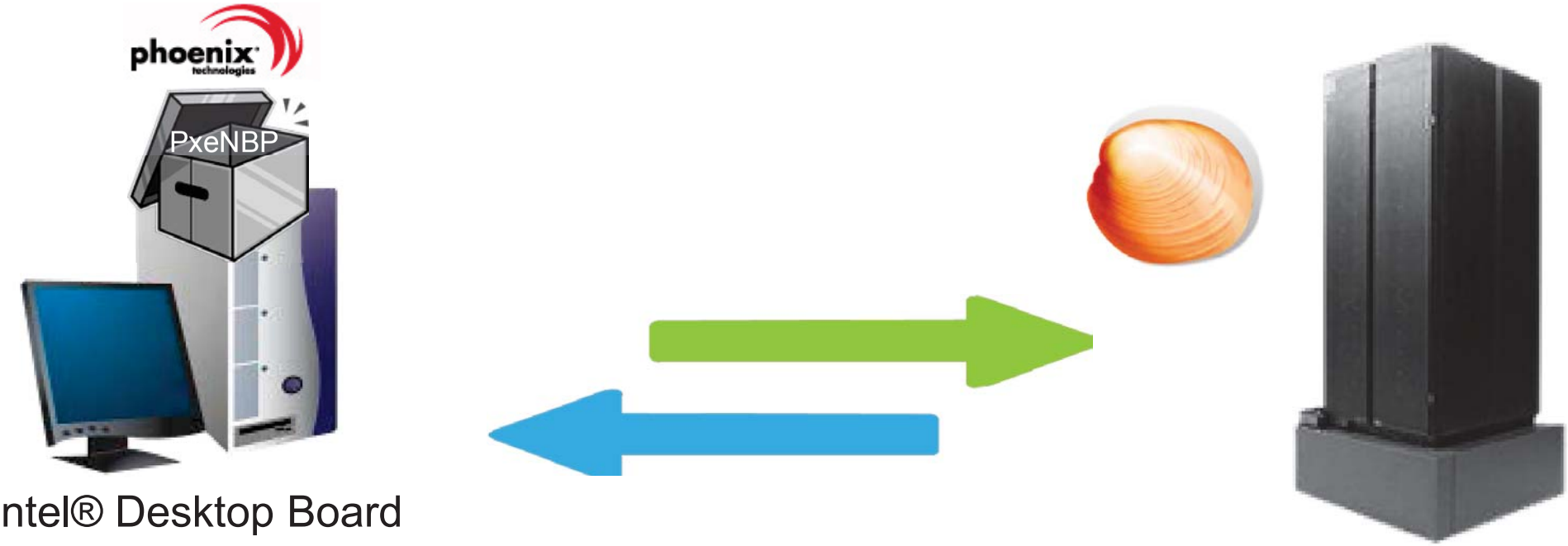
Load File Request

Intel® Desktop Board
DQ57TML

phoenix technologies

IDF2011
INTEL DEVELOPER FORUM

Send UEFI Shell



Intel® Desktop Board
DQ57TML

Run UEFI Shell

Intel® Desktop Board
DQ57TML

Demo: Remote Booting UEFI Applications & Shell Scripts

- Demo #1: Remote boot to UEFI Shell
- Demo #2: Remote boot to UEFI Shell Script
- Demo #3: Remote boot to UEFI Shell application

UEFI Manufacturing Summary

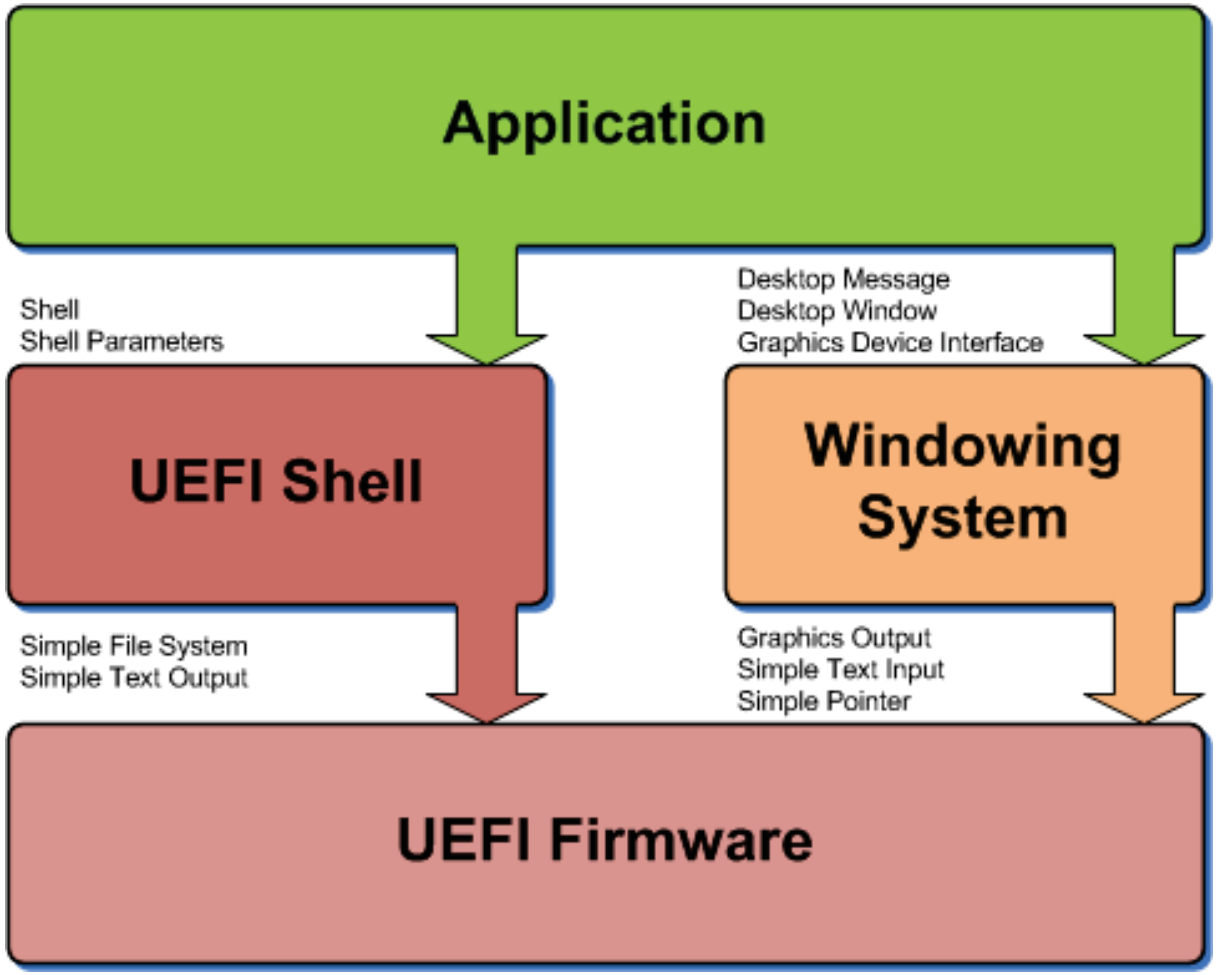
- Initialize the machine using the UEFI Shell
- Initialize the machine using server resources
- Launch the UEFI Shell from the server
- Use UEFI Shell scripts for the tasks

Agenda

- What is the UEFI Shell?
- Basic UEFI Application Programming
- UEFI Applications for Manufacturing
- UEFI Applications for Graphics

UEFI Graphics Applications

- Users expect their pre-boot experience to resemble their OS-present experience.
  - Fonts, mouse, graphics, windows.



phoenix technologies

IDF2011 INTEL DEVELOPER FORUM

phoenix technologies

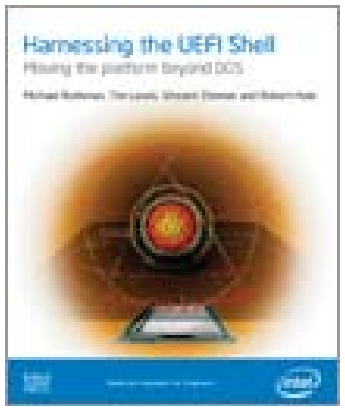
IDF2011 INTEL DEVELOPER FORUM

Summary

- The UEFI Shell is a standard, scripted, scalable command-line environment that works on any UEFI 2.1+ capable machine
- With the C Standard Library, porting over DOS applications is easy
- The UEFI Shell simplifies manufacturing using remote boot and scripting
- The UEFI Shell supports graphics application development.

Next Steps

- **Go create UEFI applications!**
- **Get the Intel® DQ57TML or Intel® DQ57TM Desktop Board from. www.tunnelmountain.com**
  - **Recommended reference platform.**
- **Get the UEFI Shell Book!**
  - ***Harnessing the UEFI Shell*, Intel Press.**
    www.intel.com/intelpress



phoenix technologies

IDF2011 INTEL DEVELOPER FORUM

Tunnel Mountain Intel DQTM57 UEFI 2.3.1 platform

Intel® UDK 2010 Compatible, supports UEFI 2.3.1

Pre-assembled systems available at HDNW, visit

http://www.Tunnelmountain.net

tomk@hdnw.com, (425) 943-5515 ext 42234. Use product name "Tunnel Mountain" when ordering

Comes with class 2 CSM and UEFI enabled firmware
Download site has Class 3 UEFI only firmware(nocsm)

Comes with serial port for debug
Can be ordered with optional ITP connector and socketed SPI flash - AC-SPEC4480

Visit http://developer.intel.com/technology/efi/uefi-ihv.htm for the latest information and other IHVs collateral

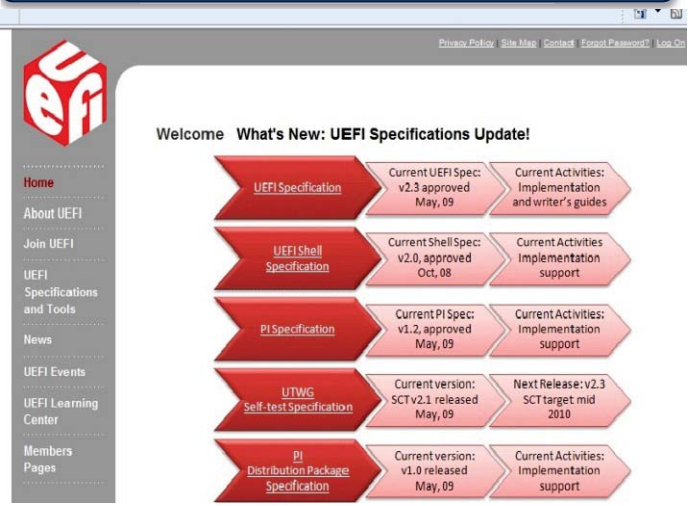
Fall 2011 UEFI Plugfest – Taipei, Oct 24-27

Visit www.UEFI.org for Event Info & Registration

IDF2011
INTEL DEVELOPER FORUM

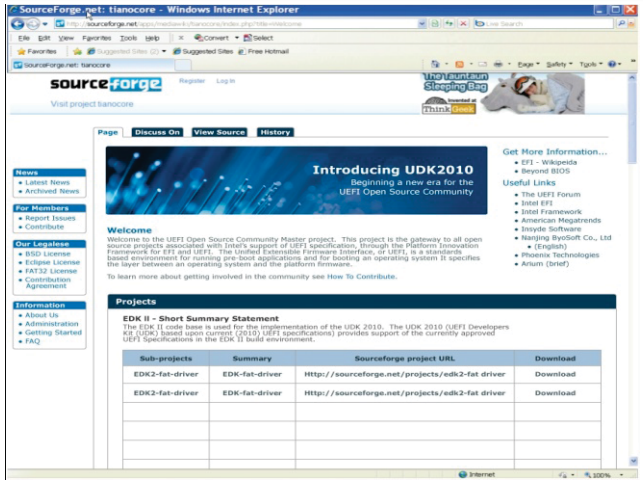
UEFI Industry Resources

UEFI Forum



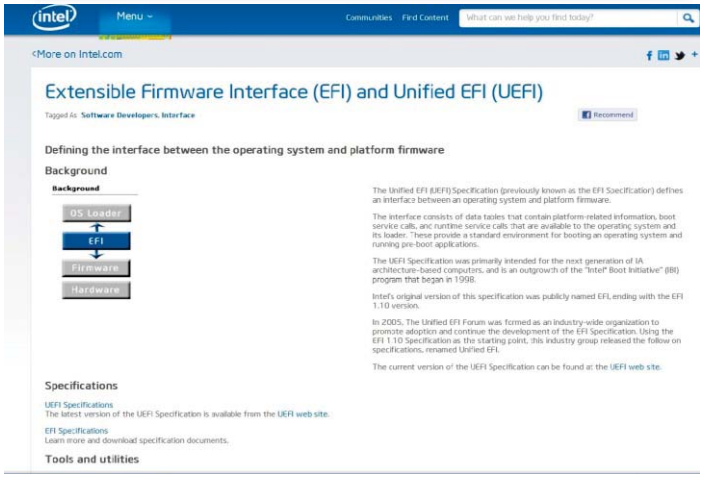
www.uefi.org

UEFI Open Source



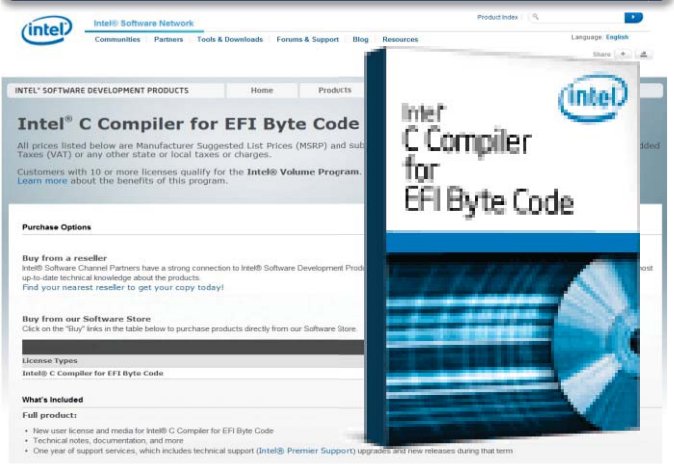
www.tianocore.org

Intel UEFI Resources



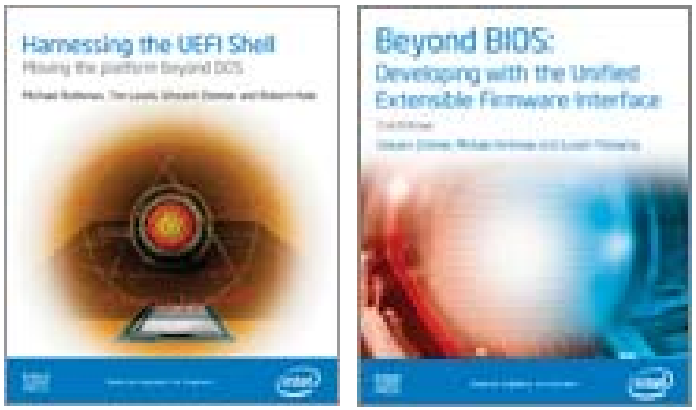
www.intel.com/technology/efi/index.htm

Intel EBC Compiler



http://software.intel.com/en-us/articles/intel-c-compiler-for-efi-byte-code-purchase/

UEFI Books



www.intel.com/intelpress

Training/IHVs Contact

Laurie Jarlstrom
- Intel UEFI Training
- Laurie.Jarlstrom@intel.com

Brian Richardson
- Intel IHVs UEFI Support
- Brian.Richardson@intel.com

UEFI Sessions Moscone SF IDF 2011

Session ID	Title	Company	Day / Time	Rm
✓ EFIS001	UEFI Security and Networking Advancements	Intel & Insyde SW	Tue 1:05 - 2:00	2009
✓ EFIS002	UEFI Innovations for Platform Security	Intel & AMI	Tue 2:10 - 3:00	2009
✓ EFIS003	Beyond DOS: UEFI Modern Pre-boot Application Development Environment	Intel & Phoenix Tech. LTD	Tue 3:20 - 4:10	2009
EFIS004	Designing for Next Generation Best-In-Class Platform Responsiveness	Intel	Tue 4:25 - 5:15	2009
EFIQ001	Hot Topic Q&A: UEFI in the Industry	All Speakers	Tue 5:25 - 6:00	2009
EFIS005	Microsoft* Windows* Platform Evolution and UEFI Requirements	Intel & Microsoft	Thu 1:05 - 1:55	2005
SPCQ003	Hot Topic Q&A: Intel & Microsoft - Windows* 8	Intel & Microsoft	Thu 2:05 - 2:55	2005

✓ = DONE

Please Fill out the Online Session Evaluation Form

Be entered to win fabulous prizes every day!

Winners will be announced at 6pm (Day 1/2) and 3:30pm (Day 3)

You will receive an email prior to the end of this session.

Q&A

Legal Disclaimer

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL® PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.
- Intel may make changes to specifications and product descriptions at any time, without notice.
- All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.
- Intel, processors, chipsets, and desktop boards 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.
- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.
- Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Go to: http://www.intel.com/products/processor_number
- Intel product plans in this presentation do not constitute Intel plan of record product roadmaps. Please contact your Intel representative to obtain Intel's current plan of record product roadmaps.
- Intel, Sponsors of Tomorrow and the Intel logo are trademarks of Intel Corporation in the United States and other countries.
- *Other names and brands may be claimed as the property of others.
-

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the second quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as “anticipates,” “expects,” “intends,” “plans,” “believes,” “seeks,” “estimates,” “may,” “will,” “should,” and their variations identify forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel’s actual results, and variances from Intel’s current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be the important factors that could cause actual results to differ materially from the company’s expectations. Demand could be different from Intel's expectations due to factors including changes in business and economic conditions, including supply constraints and other disruptions affecting customers; customer acceptance of Intel’s and competitors’ products; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Potential disruptions in the high technology supply chain resulting from the recent disaster in Japan could cause customer demand to be different from Intel’s expectations. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Revenue and the gross margin percentage are affected by the timing of Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel’s response to such actions; and Intel’s ability to respond quickly to technological developments and to incorporate new features into its products. The gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; product mix and pricing; the timing and execution of the manufacturing ramp and associated costs; start-up costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; product manufacturing quality/yields; and impairments of long-lived assets, including manufacturing, assembly/test and intangible assets. Expenses, particularly certain marketing and compensation expenses, as well as restructuring and asset impairment charges, vary depending on the level of demand for Intel's products and the level of revenue and profits. The majority of Intel’s non-marketable equity investment portfolio balance is concentrated in companies in the flash memory market segment, and declines in this market segment or changes in management’s plans with respect to Intel’s investments in this market segment could result in significant impairment charges, impacting restructuring charges as well as gains/losses on equity investments and interest and other. Intel's results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Intel’s results could be affected by the timing of closing of acquisitions and divestitures. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust and other issues, such as the litigation and regulatory matters described in Intel's SEC reports. An unfavorable ruling could include monetary damages or an injunction prohibiting us from manufacturing or selling one or more products, precluding particular business practices, impacting Intel’s ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. A detailed discussion of these and other factors that could affect Intel’s results is included in Intel’s SEC filings, including the report on Form 10-Q for the quarter ended April 2, 2011.

Rev. 5/9/11