

presented by



**American
Megatrends**



Best Practices for UEFI Driver Compatibility

**UEFI Fall Plugfest – October 24-27, 2011
Presented by Stefano Righi (AMI)**

Agenda



- Best Practices for UEFI
- UEFI Specification Evolution
- UEFI Driver Meets Platform
- User Interface: HII
- Common Driver Issues
- Debugging
- Get More Information

Best Practices for UEFI



- *Code to specification, not implementation*
- Only use UEFI standard protocols and services
 - UEFI Drivers only use Boot & Runtime Services
 - No PI, EDK or CSM protocols
- Make proper use of UEFI Human Interface Infrastructure (HII) for user interface
- Don't make legacy assumptions
- Test against multiple UEFI platforms

UEFI Specification Evolution



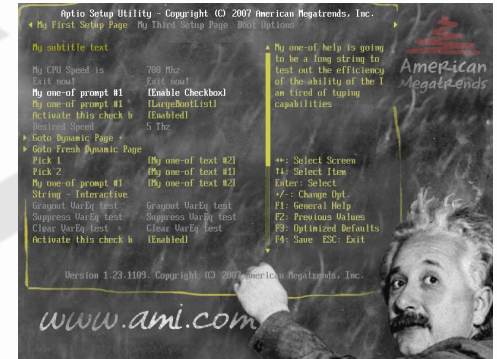
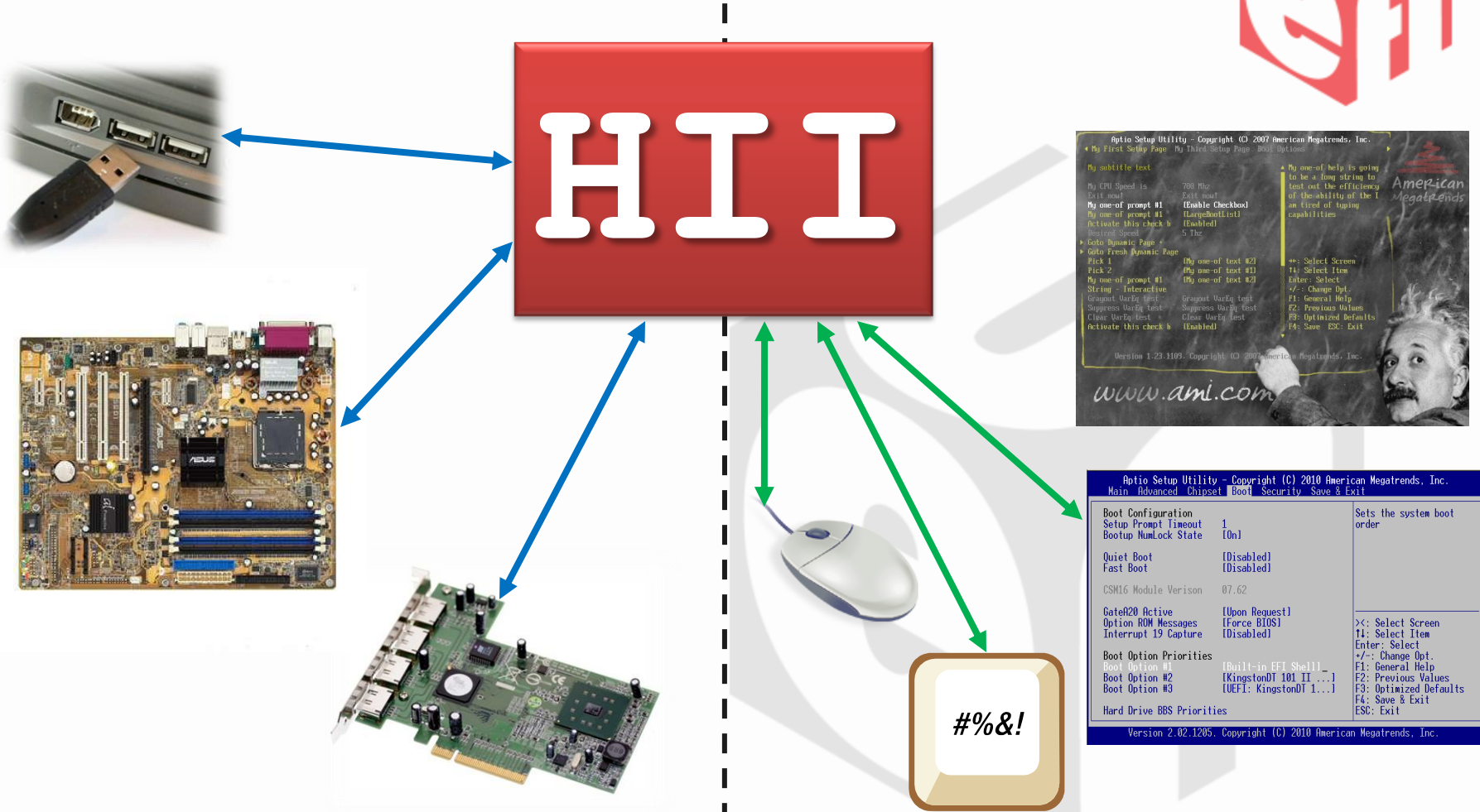
- UEFI 2.0 - Introduced UEFI Driver Model
- UEFI 2.1 – Added Protocols
 - Component Name2 Protocol
 - Driver Family Override Protocol
 - Driver Diagnostic Protocol
 - Driver Configuration & Configuration2 Protocol
- UEFI 2.2
 - Deprecated - Driver Configuration & Configuration2 Protocol
 - Added - Driver Health Protocol
- `EFI_HII_CONFIG_ACCESS_PROTOCOL.Callback()` & `EFI_BROWSER_ACTION` evolves
 - UEFI 2.1: `CHANGING`, `CHANGED`
 - UEFI 2.3: `RETIREVE`, `FORM_OPEN`, `FORM_CLOSE`
 - More to come because of pending ECRs
 - Check for `EFI_BROWSER_ACTION` and return `EFI_UNSUPPORTED` for unsupported callback types

UEFI Driver Meets Platform



- *Human Interface Infrastructure (HII)*
 - Firmware & Drivers publish to a “database”
 - Common “browser” for a consistent user interface
 - Avoid switching among different UI models
 - OEM defines the user experience
 - OEM & ODM branding happens in setup
- UEFI firmware sets platform policy for OpROM
- UEFI driver cannot assume that platform will run OpROM a certain way
- Depending on platform policy, UEFI drivers & OpROM may be loaded but not started
 - OS driver cannot assume UEFI driver has been started

User Interface: HII



Aptio Setup Utility - Copyright (C) 2010 American Megatrends, Inc.		
Main Advanced Chipset Boot Security Save & Exit		
Boot Configuration		Sets the system boot order
Setup Prompt Timeout	1	
Bootup NumLock State	[On]	
Quiet Boot	[Disabled]	
Fast Boot	[Disabled]	
CSM16 Module Version	07.62	
GateA20 Active	[Upon Request]	
Option ROM Messages	[Force BIOS]	>>: Select Screen
Interrupt 19 Capture	[Disabled]	F1: Select Item
		Enter: Select
Boot Option Priorities		+/-: Change Opt.
Boot Option #1	[Built-in EFI Shell]	F1: General Help
Boot Option #2	[KingstonDT 101 II ...]	F2: Previous Values
Boot Option #3	[UEFI: KingstonDT 1...]	F9: Optimized Defaults
		F4: Save & Exit
Hard Drive BBS Priorities		ESC: Exit
Version 2.02.1205. Copyright (C) 2010 American Megatrends, Inc.		



Questions, Data & Strings

Localization, Input & Display

User Interface: HII Tips



- Avoid direct user interaction
 - Publish protocols for firmware interaction
 - Use **DriverHealth** protocol for mandatory configuration or repair operations
- Don't directly invoke popup windows
 - Formset elements such as **InconsistentIF** can create conditions to trigger a popup
- Remember ... *drivers provide forms, the HII browser provides the user experience*
 - Look & feel varies between platforms

Common Driver Issues



- Calling non-UEFI protocols (PI & EDK)
 - **UEFI != EDK ... UEFI != PI ... UEFI != CSM**
 - *Code to the spec, not an implementation*
- Don't assume Optional protocols are available in every implementation
- Poor handling of function returns
 - Error returns & unsupported functions
- Misusing HII Database Protocols
 - Managing HII packs, generating UI elements

Common Driver Issues



- Avoid enabling unsupported PCI attributes
 - **PciIo->Attributes**
 - Support is required from the PCI Controller, PCI-to-PCI Bridge and PCI Bus Controller for an attribute to properly take effect
- Check platform attributes before enabling **EfiPciIoAttributeOperationSupported**
- Avoid using EDK macros to enable devices

Debugging



- Some drivers try directly access hardware for debug output (USB, COM, Port 80)
 - Problem: *hardware is already in use*
 - Result: the driver breaks system output
- Solution: *call standard output protocols*
 - **gST->StdErr**
 - More flexible
 - Works with new tools



Get More Information



- UEFI Forum Learning Center [\[link\]](#)
- Review the UEFI Specification ...
 - Sections 6 (Services – Boot Services)
 - Sections 7 (Services – Runtime Services)
 - Sections 9 (Protocols – Device Path)
 - Sections 10 (Protocols – UEFI Driver Model)
 - Sections 13 (PCI Bus Support)
 - Sections 20 (EFI Byte Code Virtual Machine)

Thanks for attending the
UEFI Fall Plugfest 2011



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

presented by



**American
Megatrends**



But wait, there's more ...

T



Welcoming Remarks – Aven Chuang, Insyde Software
UEFI Forum Updates – Dong Wei, VP of the UEFI Forum

T



Best Practices for UEFI Driver Compatibility – Stefano Righi, American Megatrends, Inc.
Understanding Platform Requirements for UEFI HII – Brian Richardson, Intel Corporation

W



UEFI Security Enhancements – Kevin Davis, Insyde Software
How to Protect the Pre-OS Environment with UEFI – Tony Mangefeste, Microsoft

Th



Pre-OS Display Switching using GOP – James Huang, AMD
Debug Methodology Under UEFI – Jack Wang, Phoenix Technologies

Download presentations after the plugfest at www.uefi.org