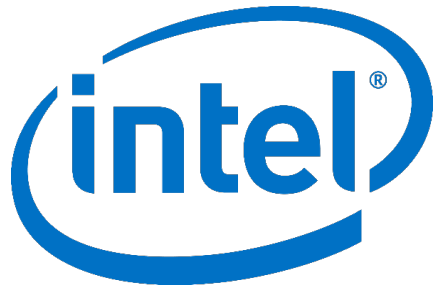


presented by



Code Coverage in Firmware Automation Testing

Spring 2017 UEFI Seminar and Plugfest
March 27 - 31, 2017

Presented by Liu Zhi (Intel Corporation)

Agenda



- Introduction
- Code Coverage
- Example Code Coverage Implementation w/ EDK II
- Tips & Tricks
- Summary / Q&A

Introduction



- Test automation is common in firmware environments
- Typically automation is designed to replace tests with human interaction
- However, automation can cover a larger set of scenarios for firmware

Firmware Test Challenges



- Simulating user input in automation
 - Presented as USB HID (keyboard/mouse)
- Adjusting test cases to output
 - LED status, screen output, text recognition, etc.
- Persistence across platform reset
- **Determining coverage of validation plans**
- This session will focus on the last item

Agenda



- Introduction
- **Code Coverage**
- Example Code Coverage Implementation w/ EDK II
- Tips & Tricks
- Summary / Q&A

Code Coverage



- Code coverage is a measure of the degree source code is tested by a test suite
 - Identifies what areas of the code are exercised during program execution
 - High code coverage = more thorough testing
- Code coverage reports direct users to add tests for uncovered code & find dead code
- Best with a full automation solution

Code Coverage in Firmware



- Code coverage is commonly used in application, but not as much in firmware
- Firmware use cases typically require changes in configuration
 - Change setup menu options
 - Add/remove peripherals (USB, SATA, ...)
- Implement without source changes

Metrics for Code Coverage























- Measure percentages for three specific types of execution coverage:
 - Functions
 - Conditions
 - Lines
- Metrics can be general (entire firmware project) or granular (focus on one file)

Example Report



Functions:	233	191	81%
Branches Coverage:	[0%-15%]	[15%-50%]	[50%-100%]
Coverage Color:			
Functions Coverage:	[0%-15%]	[15%-50%]	[50%-100%]
Coverage Color:			

Directory	Branches		Functions	
s:/MdeModulePkg/Core/Dxe/Dispatcher/		32 % 143/444		81 % 18/22
s:/MdeModulePkg/Core/Dxe/DxeMain/		34 % 77/222		68 % 11/16
s:/MdeModulePkg/Core/Dxe/Event/		67 % 153/228		100 % 22/22
s:/MdeModulePkg/Core/Dxe/FwVol/		46 % 116/249		76 % 16/21
s:/MdeModulePkg/Core/Dxe/FwVolBlock/		36 % 28/76		36 % 4/11
s:/MdeModulePkg/Core/Dxe/Gcd/		44 % 291/655		84 % 33/39
s:/MdeModulePkg/Core/Dxe/Hand/		56 % 453/798		100 % 36/36
s:/MdeModulePkg/Core/Dxe/Image/		37 % 155/411		60 % 9/15
s:/MdeModulePkg/Core/Dxe/Library/		50 % 13/26		100 % 3/3
s:/MdeModulePkg/Core/Dxe/Mem/		47 % 284/604		80 % 20/25

Agenda



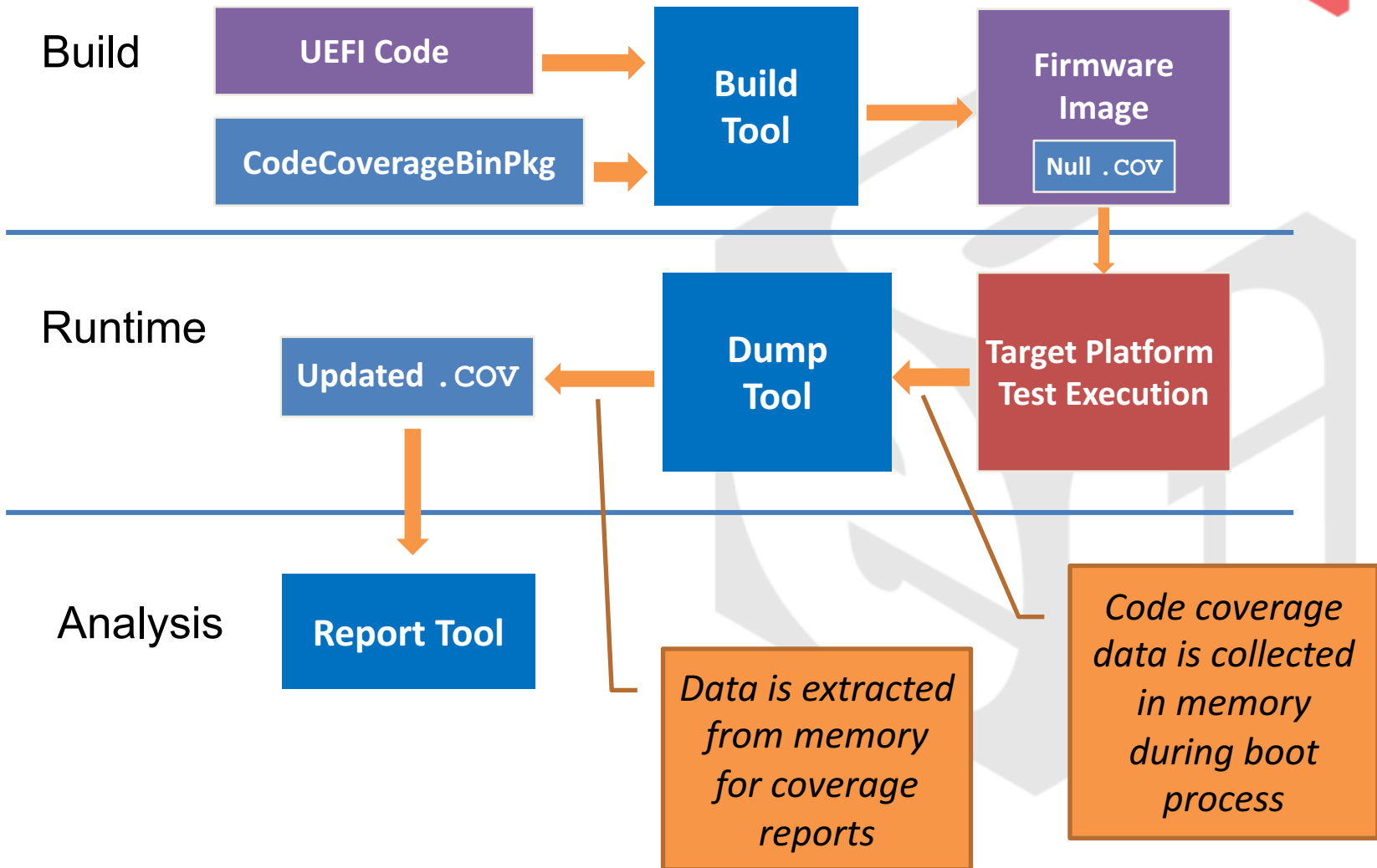
- Introduction
- Code Coverage
- **Example Code Coverage Implementation w/ EDK II**
- Tips & Tricks
- Summary / Q&A

Example Code Coverage Implementation w/ EDK II

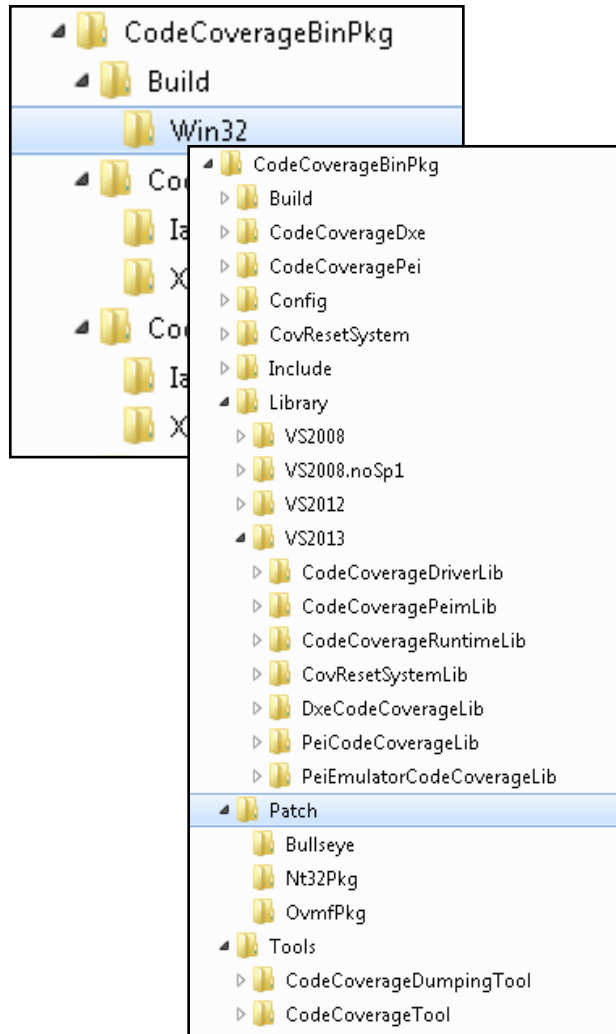


- This example is based on an EDK II project from tianocore.org
- Test automation is implemented using [Intel® Intelligent Test System](#) (Intel® ITS)
- Code coverage is an add-on package for Intel ITS, using [Bullseye](#)

Overall Workflow



Configuring Code Coverage



- Add **CodeCoverageBinPkg**
- Patch EDK II build tools
 - **Build\Win32**
 - Removes need to modify code
- Edit platform config files to add service drivers (PEI, DXE & Reset)
- Contains additional tools
 - Libraries for Visual Studio
 - Patches for Bullseye & two EDK II reference platforms (NT32, OVMF)
 - Tools for report generation



Build with Code Coverage



- Build project to enable code coverage

- Example for NT32 (UEFI emulator)

```
subst s: c:\Edk2
s:\> CodeCoverageBinPkg\Build\Win32\CovEdkSetup.bat
s:\> set ITSCOVTOOLCHAIN=VS2013
s:\> build -t VS2013x86 -E
CodeCoverageBinPkg\Config\Nt32Pkg.ini
-D ENCOV -D ITSCOV_VS2013_ENABLE=TRUE -Z
```

- Boot platform to execute test plan and gather code coverage data

Dump Code Coverage Data



```
Shell> fsnt0:
fsnt0:\> ucovdump.efi -?

ITS Code Coverage Dumping Tool (ICCDT), Version 0.4
This tool enables you to dump the code coverage data in EDKII platf

Usage:
  ICCDT [-o <OutFn>]
  ICCDT [-z]
  ICCDT [-?]

Options:
  <InFn>  The new code coverage file name to be updated.
  <OutFn>  The output code coverage file name.
  -o      Dump the code coverage data from memory to current media
  -z      Clean the code coverage data in memory.
  -?      Help information.

fsnt0:\> ucovdump.efi -o Edk2CovData.cov
CovFileBase = 0x62B5240, CovFileSize = 0x6F3E2
Congratulations! The Edk2CovData.cov has been created in current pa

fsnt0:\> _
```

Code coverage data is collected in memory during boot & dumped under one of three conditions ...

1. Collected from UEFI Shell (**ucovdump.efi**)
2. Stored to USB drive on system reset
3. Stored to USB drive at **ExitBootServices ()**
(not supported in NT32)

Report Generation



ucovreport creates coverage reports

```
ucovreport report_html -o report --ucno CodeCoverageBinPkg\Config\Edk2Cov.ucno --ucda e:\Edk2CovData.cov
```

Use these results to find testing gaps

Please see the details below:

ITS Code Coverage Report

Current view: [index - z:/](#)

Test: [edk2cov.ucno itscov_0530_000014.cov](#)

Date: 2016-5-30

	Found	Hit	Coverage
Branches:	4562	145	3%
Functions:	571	29	5%
Branches Coverage:	[0%-15%)	[15%-50%)	[50%-100%)
Coverage Color:			
Functions Coverage:	[0%-15%)	[15%-50%)	[50%-100%)
Coverage Color:			

Directory	Branches	Functions
z:/MdeModulePkg/Universal/SmbiosDxe/	31 % 117/372	84 % 11/13
z:/MdePkg/Library/	0 % 28/4190	3 % 18/558

Agenda



- Introduction
- Code Coverage
- Example Code Coverage Implementation w/ EDK II
- **Tips & Tricks**
- Summary / Q&A

Tips & Tricks



- Based on current implementation of Intel® ITS using Bullseye
 - Boot speed when code coverage is enabled
 - Build issues integrating code coverage
 - Code coverage in SEC & PEI phases
 - Changes in firmware image size
 - Code coverage for assembly code

Tips & Tricks



- Boot speed when code coverage is enabled
 - There is some overhead for code coverage entry/exit routines (increases boot time)
 - Intended for test environments only
- Build issues integrating code coverage
 - Make sure you are using the patched build tools (**build.exe** and **GenFds.exe**)
 - Note base tools version for your compiler will differ from EDK II master tree version

Tips & Tricks



- Code coverage in SEC & PEI phases
 - Not supported at this time
- Changes in firmware image size
 - Compiled image size will be increased, due to additional support for code coverage
- Code coverage for assembly code
 - Not supported at this time

Agenda



- Introduction
- Code Coverage
- Example Code Coverage Implementation w/ EDK II
- Tips & Tricks
- **Summary / Q&A**

Summary



- Test automation is an important method for validating firmware solutions
- Code coverage should be applied to firmware, as well as application code
- Analysis helps identify dead code and untested code functions/branches
- Code coverage solutions are available for UEFI and EDK II



Q&A



Thanks for attending the
Spring 2017 UEFI Seminar
and Plugfest



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



presented by

