Learn How to Build Linux in a Virtual Environment with the **Linux SDK from Freescale**

FTF-DES-F1163

Dr. Xin-Xin Yang | Software Engineering Manager

J U N E . 2 0 1 5

freescale™

Agenda

- Introduction to Freescale QorIQ Linux SDK

- OOBE Issues and Why Virtual Environment

- Linux SDK Virtual Machine Environment

  – Key Content

  – Deployment Process

  – Usage and Application

  – Release and Distribution

- Summary

- Q&A

# Freescale QorIQ Linux SDK Introduction

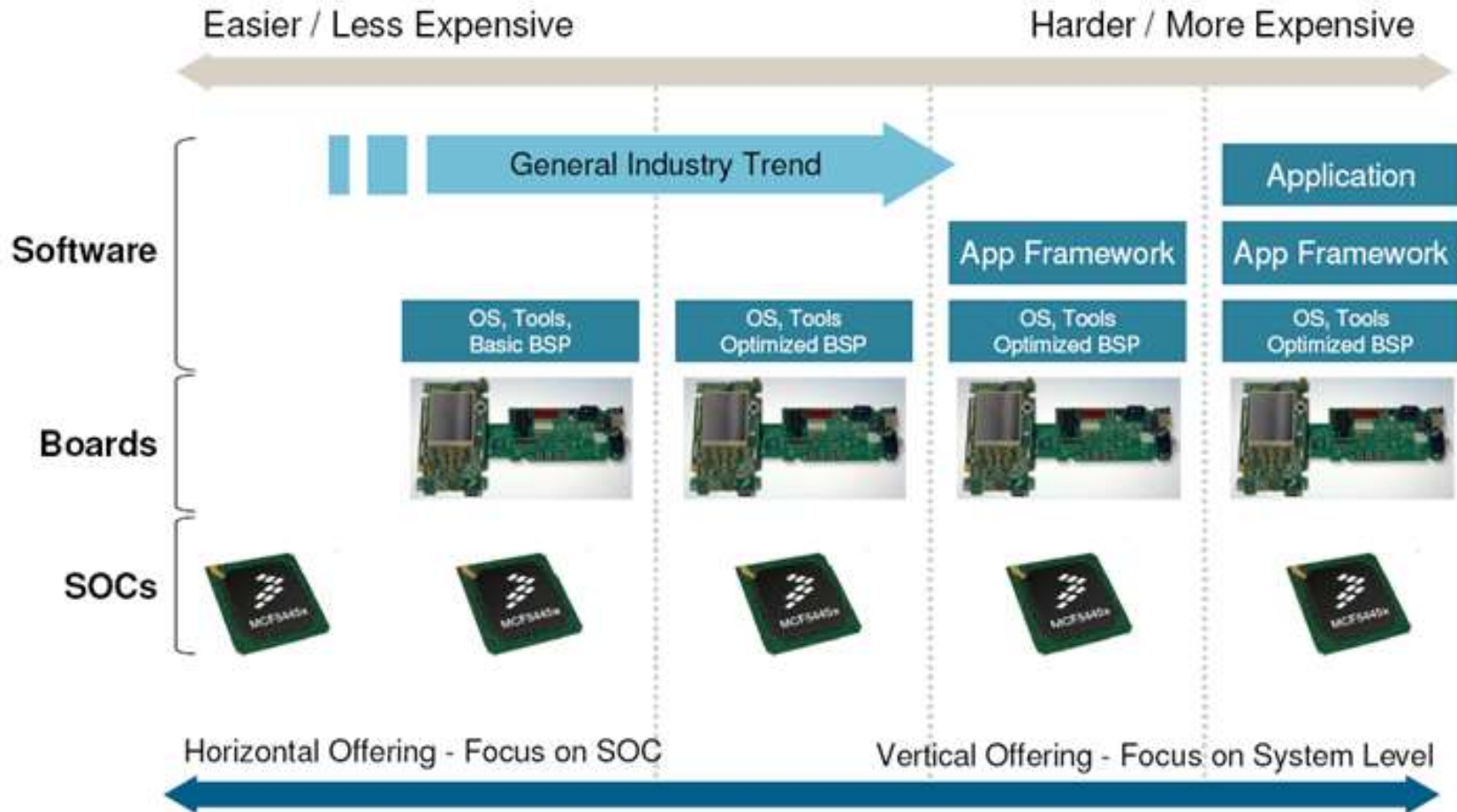# Software and Enablement: Technical Strategy

- **Optimized Platforms**
  - Silicon-optimized software IP across all of our hardware devices and platforms
  - Efficient cores and peripheral initialization and device driver generation
  - Stand-alone base tools and run-time technologies
  - Multicore, accelerators, DSP support

- **Vertical Solutions**
  - Complete solutions in select application spaces
  - Built around standard platforms
  - Available throughout the ecosystem
    - Deeper partnerships
  - Application portability across silicon platforms and product generations
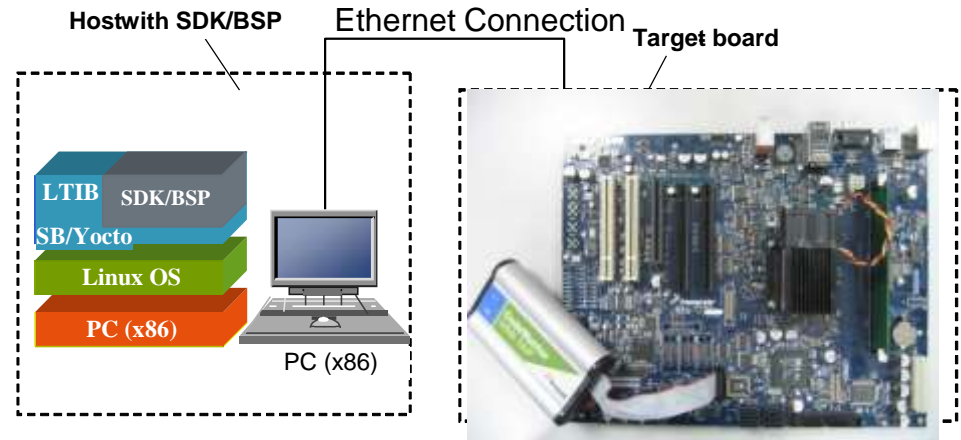
#FTF2015
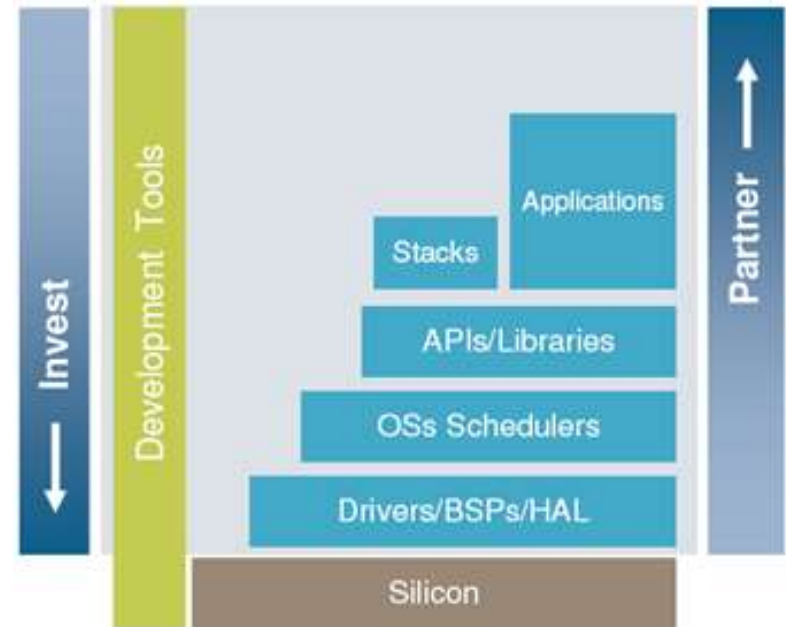
Marketing Trends: Software Solutions

#FTF2015

Distributing Linux: Four Primary Models

Model	Approach	When to Use	Attributes
Native on Eval Board	Provide evaluation boards with complete native GNU tool environments right on the board.	Desire zero "getting started" effort in building and running Freescale solutions and standard OSS	Easy to use
BSP/SDK	This embedded distribution helps customers create an entire Linux system, package both as an ISO image and also in a virtual machine.	Need a tool to generate a complete Linux environment including tailored file system.	Comprehensive, but very flexible and powerful
A la Carte	Simplifies customer access to major Freescale-created Linux components. Perfect for integration into Linux distributions from other sources, home-brew or 3rd party. Supports fast delivery of patches.	Desire to integrate Freescale Linux components into a Linux development environment that the customer already uses	Simple when the customer is also the integrator
Opensource	Committed all the patches to opensource community and push to get them upstreamed. Users directly access the opensource to build.	Desire to use software purely from the community	Simple and flexible

freescale™

#FTF2015

QorIQ Linux SDK Based on Yocto

- What is an SDK/BSP?
  - Software Development Kit
  - Board Support Package
- What does an SDK include?
  - Boot loader
  - Kernel
  - Tool chain
  - File system
  - RAM disk
  - NFS
  - Hard disk (ghost image)
  - Applications
  - Deployment mechanism
  - Documentation
- Where can you get SDKs?
  - External users
  - http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=SDKLINUXDPAA
  - http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=SDKLINUX&parentCode=null&nodeId=0152100332BF69

#FTF2015

Freescale Linux SDK/BSP

- Linux Software Development Kit / Board Support Packages for Freescale silicon are tested, certified and frozen, ensuring a fully operational tool chain, kernel and board-specific modules that are ready to use together within a fixed configuration for specific hardware reference platforms.

- Freescale-developed Linux SDK/BSPs provide customers with a comprehensive starting point for their Linux development efforts on a variety of platforms
  - QorIQ processors
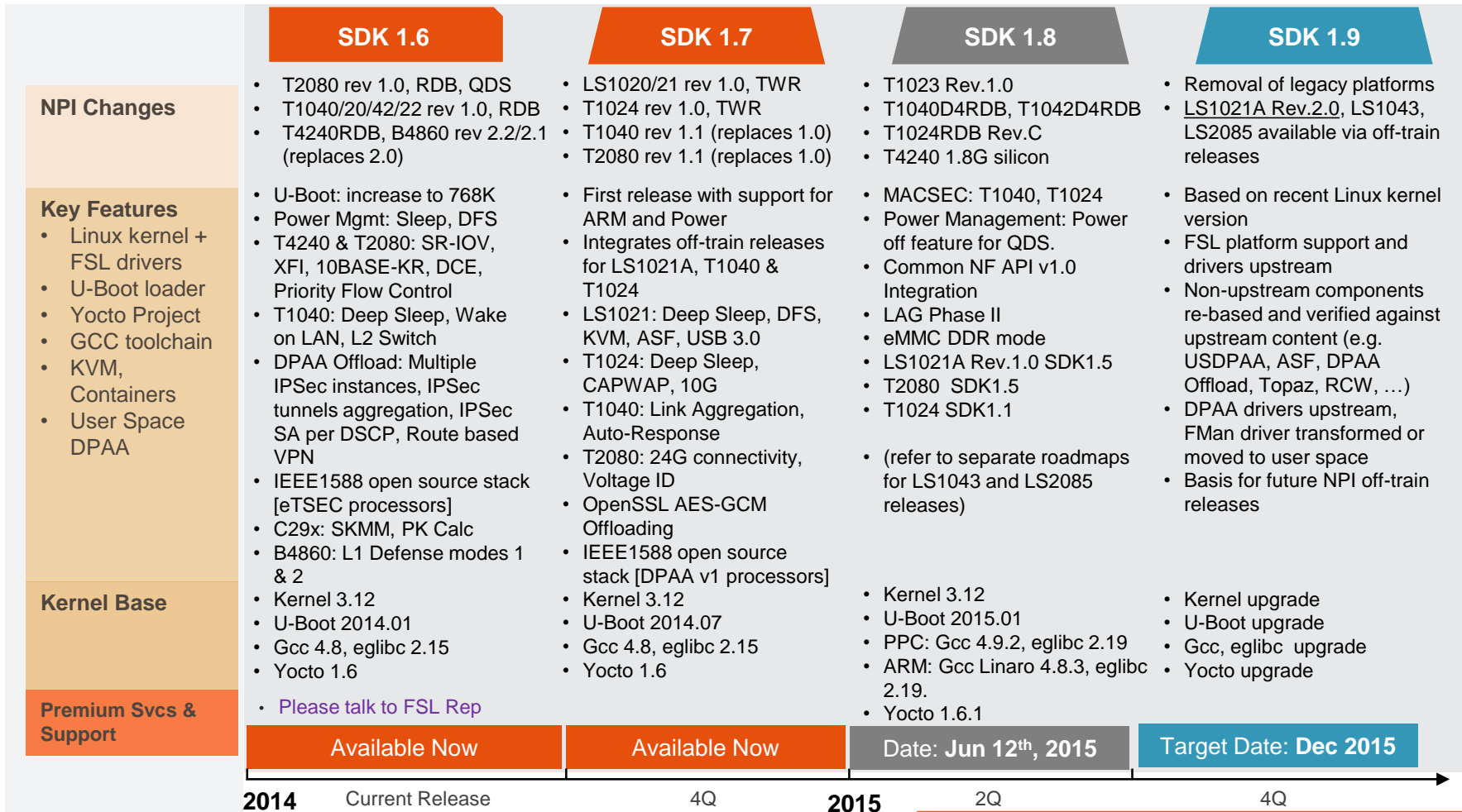 - QorIQ LS series processors

Key Components in QorIQ Linux SDK

- Yocto
- GNU Toolchain
- U-Boot Boot Loader
- Linux Kernel and Virtualization
- Linux Kernel and Device Drivers
- Application Specific Fastpath (ASF)
- User Space Datapath Acceleration Architecture (USDPAA) and Applications
- Freescale Embedded Hypervisor (Topaz)
- Other Tools and Utilities

#FTF2015

QorIQ Linux SDK Roadmap

	SDK 1.6	SDK 1.7	SDK 1.8	SDK 1.9
NPI Changes	• T2080 rev 1.0, RDB, QDS • T1040/20/42/22 rev 1.0, RDB • T4240RDB, B4860 rev 2.2/2.1 (replaces 2.0)	• LS1020/21 rev 1.0, TWR • T1024 rev 1.0, TWR • T1040 rev 1.1 (replaces 1.0) • T2080 rev 1.1 (replaces 1.0)	• T1023 Rev.1.0 • T1040D4RDB, T1042D4RDB • T1024RDB Rev.C • T4240 1.8G silicon	• Removal of legacy platforms • <u>LS1021A Rev.2.0</u>, LS1043, LS2085 available via off-train releases
Key Features • Linux kernel + FSL drivers • U-Boot loader • Yocto Project • GCC toolchain • KVM, Containers • User Space DPAA	• U-Boot: increase to 768K • Power Mgmt: Sleep, DFS • T4240 & T2080: SR-IOV, XFI, 10BASE-KR, DCE, Priority Flow Control • T1040: Deep Sleep, Wake on LAN, L2 Switch • DPAA Offload: Multiple IPSec instances, IPSec tunnels aggregation, IPSec SA per DSCP, Route based VPN • IEEE1588 open source stack [eTSEC processors] • C29x: SKMM, PK Calc • B4860: L1 Defense modes 1 & 2	• First release with support for ARM and Power • Integrates off-train releases for LS1021A, T1040 & T1024 • LS1021: Deep Sleep, DFS, KVM, ASF, USB 3.0 • T1024: Deep Sleep, CAPWAP, 10G • T1040: Link Aggregation, Auto-Response • T2080: 24G connectivity, Voltage ID • OpenSSL AES-GCM Offloading • IEEE1588 open source stack [DPAA v1 processors]	• MACSEC: T1040, T1024 • Power Management: Power off feature for QDS. • Common NF API v1.0 Integration • LAG Phase II • eMMC DDR mode • LS1021A Rev.1.0 SDK1.5 • T2080 SDK1.5 • T1024 SDK1.1 • (refer to separate roadmaps for LS1043 and LS2085 releases)	• Based on recent Linux kernel version • FSL platform support and drivers upstream • Non-upstream components re-based and verified against upstream content (e.g. USDPAA, ASF, DPAA Offload, Topaz, RCW, …) • DPAA drivers upstream, FMan driver transformed or moved to user space • Basis for future NPI off-train releases
Kernel Base	• Kernel 3.12 • U-Boot 2014.01 • Gcc 4.8, eglibc 2.15 • Yocto 1.6	• Kernel 3.12 • U-Boot 2014.07 • Gcc 4.8, eglibc 2.15 • Yocto 1.6	• Kernel 3.12 • U-Boot 2015.01 • PPC: Gcc 4.9.2, eglibc 2.19 • ARM: Gcc Linaro 4.8.3, eglibc 2.19. • Yocto 1.6.1	• Kernel upgrade • U-Boot upgrade • Gcc, eglibc upgrade • Yocto upgrade
Premium Svcs & Support	• Please talk to FSL Rep			
	Available Now	**Available Now**	Date: **Jun 12th, 2015**	Target Date: **Dec 2015**

2014 Current Release 4Q **2015** 2Q 4Q

Color Legend

| | Released | | Roadmap Date | Current Release |
| | Committed | | Changes | Major Release |

#FTF2015

Supported Target: QorIQ Value-Performance Processors - I

Processor	Board	SDK 1.2	SDK 1.3	SDK 1.3.2	SDK 1.4	SDK 1.5	SDK 1.6	SDK 1.7
LS1021/20A		N	N	N	N	N	N	rev 1.0
	TWR-LS1021A	N	N	N	N	N	N	✔
P1010/P1014		rev 1.0	rev 1.0	rev 1.0	rev 1.0	rev 2.0	rev 2.01	rev 2.01
	P1010RDB	✔	✔	✔	✔	N	N	N
	P1010RDB-PB	N	N	N	N	✔	✔	✔
P1020/P1011		rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1
	P1020RDB-PC	✔	✔	✔	N	N	N	N
	P1020RDB-PD	N	N	N	✔	✔	✔	✔
P1021/P1012		rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1
	P1021RDB-PC	✔	✔	✔	✔	✔	✔	✔
P1022/P1013		rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1
	P1022DS-PA	✔	✔	✔	N	N	N	N

Supported Target: QorIQ Value-Performance Processors- II

	P1022DS-PB	N	N	N	✔	✔	✔	✔
P1023/P1017		rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1
	P1023RDB-PA	N	N	N	✔	✔	✔	✔
	P1023RDS	✔	✔	✔	N	N	N	N
P1024/P1015		rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1
	P1024RDB-PC	✔	✔	✔	N	N	N	N
P1025/P1016		rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1	rev 1.1
	P1025RDB-PC	✔	✔	✔	N	N	N	N
	TWR-P1025	N	N	N	✔	✔	✔	✔
T1024/14		N	N	N	N	N	N	rev 1.0
	T1024RDB	N	N	N	N	N	N	✔
T1040		N	N	N	N	N	rev 1.0	rev 1.1
	T1040RDB	N	N	N	N	N	✔	✔
T1042		N	N	N	N	N	rev 1.0	rev 1.1
	T1042RDB	N	N	N	N	N	✔	✔

Supported Target: QorIQ Mid-Performance Processors

Processor	Board	SDK 1.2	SDK 1.3	SDK 1.3.2	SDK 1.4	SDK 1.5	SDK 1.6	SDK 1.7
P2020		rev 2.1	rev 2.1	rev 2.1	rev 2.1	rev 2.1	rev 2.1	rev 2.1
	P2020RDB-PCA	✔	✔	✔	✔	✔	✔	✔
P2040/P2041		rev 1.0	rev 1.0 rev 1.1	rev 1.1 rev 2.0	rev 1.1 rev 2.0	rev 2.0	rev 2.0	rev 2.0
	P2041RDB-PA	✔	✔	N	N	N	N	N
	P2041RDB-PB	N	✔	✔	✔	N	N	N
	P2041RDB-PC	N	N	✔	✔	✔	✔	✔
P3041		rev 1.0 rev 1.1	rev 1.1	rev 1.1 rev 2.0	rev 1.1 rev 2.0	rev 2.0	rev 2.0	rev 2.0
	P3041DS-PA	✔	✔	✔	✔	N	N	N
	P3041DS-PC	✔	✔	✔	✔	✔	✔	✔
T2080		N	N	N	N	N	rev 1.0	rev 1.1
	T2080QDS	N	N	N	N	N	✔	✔
	T2080RDB/PCIE	N	N	N	N	N	✔	✔

freescale™

#FTF2015

Supported Target: QorIQ High-Performance Processors

Processor	Board	SDK 1.2	SDK 1.3	SDK 1.3.2	SDK 1.4	SDK 1.5	SDK 1.6	SDK 1.7
P4080		rev 2.0	rev 2.0	rev 2.0	rev 2.0	rev 3.0	rev 3.0	rev 3.0
	P4080DS	✔	✔	✔	✔	✔	✔	✔
P5020/P5010		rev 1.0	rev 2.0	rev 2.0	rev 2.0	rev 2.0	rev 2.0	rev 2.0
	P5020DS	✔	✔	✔	✔	✔	✔	✔
P5040/P5021		rev 1.0	rev 1.0	rev 2.0	rev 2.0	rev 2.1	rev 2.1	rev 2.1
	P5040DS	✔	✔	✔	✔	✔	✔	✔
T4240/T4080/T4160		N	N	rev 1.0	rev 1.0	rev 2.0	rev 2.0	rev 2.0
	T4240QDS	N	N	✔	✔	✔	✔	✔
	T4240RDB	N	N	N	N	N	✔	✔
	T4080RDB	N	N	N	N	N	✔	✔

Supported Target: QorIQ C29x Crypto Processors

Processor	Board	SDK 1.2	SDK 1.3	SDK 1.3.2	SDK 1.4	SDK 1.5	SDK 1.6	SDK 1.7
C29x		N	N	N	rev 1.0	rev 1.0	rev 1.0	rev 1.0
	C29xPCIE	N	N	N	✔	✔	✔	✔

Supported Target: QorIQ Qonverge Processors

Processor	Board	SDK 1.2	SDK 1.3	SDK 1.3.2	SDK 1.4	SDK 1.5	SDK 1.6	SDK 1.7
BSC9131		N	N	N	rev 1.0	rev 1.1	rev 1.1	rev 1.1
	BSC9131RDB	N	N	N	✔	✔	✔	✔
BSC9132		N	N	N	rev 1.0	rev 1.1	rev 1.1	rev 1.1
	BSC9132QDS	N	N	N	✔	✔	✔	✔
B4420		N	N	rev 1.0	rev 1.0	rev 1.0 rev 2.1	rev 2.2	rev 2.2
	B4420QDS	N	N	✔	✔	✔	✔	✔
B4860/B4460		N	N	rev 1.0	rev 1.0	rev 1.0 rev 2.1	rev 2.2	rev 2.2
	B4860QDS	N	N	✔	✔	✔	✔	✔

freescale™

#FTF2015

Supported Target: QorIQ PowerQUICC III Processors

Processor	Board	SDK 1.2	SDK 1.3	SDK 1.3.2	SDK 1.4	SDK 1.5	SDK 1.6	SDK 1.7
MPC8536		rev 1.2	rev 1.2	rev 1.2	N	N	N	N
	MPC8536DS	✔	✔	✔	N	N	N	N
MPC8544		rev 2.1	rev 2.1	rev 2.1	N	N	N	N
	MPC8544DS	✔	✔	✔	N	N	N	N
MPC8548		rev 3.1	rev 3.1	rev 3.1	N	N	N	N
	MPC8548DS	✔	✔	✔	N	N	N	N
MPC8572		rev 2.2.1	rev 2.2.1	rev 2.2.1	N	N	N	N
	MPC8572DS	✔	✔	✔	N	N	N	N

Feature Supported Matrix - I

Feature	P1010	P1020	P1021	P1022	P1023	TWR-P1025	P2020	P2041	P3041	P4080	P5020	P5040	T1024	T1040	T1042	T2080	T4240	C29x	BSC9131	BSC9132	B4860	B4420	LS1021A
	SDK 1.7; Yocto 1.6.1; Linux 3.12; GCC4.8.1; U-Boot 2014-07																						
Si/Board	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
32b apps	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
64b apps	na	na	na	na	na	na	na	na	na	na	Y	Y	Y	Y	Y	Y	Y	na	na	na	Y	Y	na
36b phys mem	Y	Y	Y	Y	na		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na	na	Y	Y	Y
Huge Pages (tlbfs)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y
Real-time						Y				Y											Y	Y	
Multithreading	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	Y	Y	na	na	na	Y	Y	na
Secure Boot	Y		na	na	na	na		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y
KVM	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y				Y	Y	Y
LXC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y				Y	Y	Y
libvirt	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y				Y	Y	Y
Topaz	na	na	na	na	na	na	na	Y	Y	Y	Y	Y	Y	Y		Y	Y		na	na	Y	Y	na
Linux	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
USDPAA	na	na	na	na		na	na	Y	Y	Y	Y	Y	Y	Y		Y	Y	na	na	na	Y	Y	na
ASF	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y		Y	Y	Y	Y		Y	Y			Y
OpenSSL	Y				Y			Y		Y	Y	Y	Y	Y	Y	Y	Y	Y			Y	Y	na

Feature Supported Matrix - II

Feature	P1010	P1020	P1021	P1022	P1023	TWR-P1025	P2020	P2041	P3041	P4080	P5020	P5040	T1024	T1040	T1042	T2080	T4240	C29x	BSC9131	BSC9132	B4860	B4420	LS1021A
						SDK 1.7; Yocto 1.6.1; Linux 3.12; GCC4.8.1; U-Boot 2014-07																	
L1 Defense	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na			Y	Y	na
DSP Boot	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na			Y	Y	na
DSP IPC	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na			Y	Y	na
Linux IPFwd	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y
Linux IPSec	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y
Linux Termination	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y
Linux NAS	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na					
Linux RAID	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na					
Linux SATA	Y	Y	Y	Y	na	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na					
ASF IPFwd	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na	Y	Y			Y
ASF IPSec	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na	Y	Y			Y
USDPAA IPFwd	na	na	na	na		na	na	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na	na	na	Y	Y	na
USDPAA IPSec	na	na	na	na		na	na	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na	na	na	Y	Y	na
USDPAA Reflector	na	na	na	na		na	na	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na	na	na	Y	Y	na

#FTF2015

Feature Supported Matrix - III

Feature	P1010	P1020	P1021	P1022	P1023	TWR-P1025	P2020	P2041	P3041	P4080	P5020	P5040	T1024	T1040	T1042	T2080	T4240	C29x	BSC9131	BSC9132	B4860	B4420	LS1021A
	SDK 1.7; Yocto 1.6.1; Linux 3.12; GCC4.8.1; U-Boot 2014-07																						
Audio	na	na	na	Y	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	Y
DCE	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	Y	Y	na	na	na	na	na	na
DIU/Video	na	na	na	Y	na	na	na	na	na	na	na	na	Y	Y	Y	na	na	na	na	na	na	na	Y
DPAA Offload	na	na	na	na		na	na	Y		Y						Y	Y	na	na	na	Y	Y	na
FlexCAN	Y	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na				na	na	Y
FMAN/BMAN/QMAN	na	na	na	na	Y	na	na	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na	na	na	Y	Y	na
I2C	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
IEEE1588, basic	Y	Y	Y	Y		Y	Y			Y			Y	Y		Y			Y		Y	Y	
IEEE1588, IXXAT																Y			Y		Y	Y	
IFC	Y	na	na	na	na	na	na	na	na	na	na	na	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
L2 Switch	na	na	na	na	na	na	na	na	na	na	na	na	Y	na	na	na	na	na	na	na	na	na	na
QE	na	na	Y	na	na	Y	na	na	na	na	na	na	Y	Y	Y	na	na	na	na	na	na	na	na
PAMU	na	na	na	na	na	na	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na	na	na	Y	Y	na
PCIe RC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y				Y	Y	Y
PCIe EP										Y						Y	Y	Y					
PME	na	na	na	na	na	na	na	Y	Y	Y	Y	na	na	Y	Y	Y	Y	na	na	na	Y	Y	na
Power Management	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y

#FTF2015

Feature Supported - IV

Feature	P1010	P1020	P1021	P1022	P1023	TWR-P1025	P2020	P2041	P3041	P4080	P5020	P5040	T1024	T1040	T1042	T2080	T4240	C29x	BSC9131	BSC9132	B4860	B4420	LS1021A
	SDK 1.7; Yocto 1.6.1; Linux 3.12; GCC4.8.1; U-Boot 2014-07																						
RAID HW assist	na	na	na	na	na	na	na	na	na	na	Y	Y	na	na	na	na	na	na	na	na	na	na	na
RMAN	na	na	na	na	na	na	na	Y	Y	Y	Y	na	na	na	na	Y	Y	na	na	na	Y	Y	na
SATA	Y	na	na	Y	na	na	na	Y	Y	na	Y	Y	Y	Y	Y	Y	Y	na	na	na	na	na	
SDHC		Y	Y	Y	na	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na	Y	na	na	Y
SEC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SPI	Y	Y	Y	Y	Y	na	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SRIO	na	na	na	na	na	na	na	Y	Y	Y	Y	na	na	na	na	Y	Y	na	na	na	Y	Y	na
TDM	Y	Y		Y	na	na	na	na	na	na	na	na	Y	Y	Y	na	na	na	na	na	na	na	na
USB	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	na	Y	Y	Y	Y	Y
veTSEC/e TSEC/FEC	Y	Y	Y	Y	na	Y	Y	na	na	na	na	na	na	na	na	na	na	Y	Y	Y	na	na	Y
Watchdog	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y
XAUI	na	na	na	na	na	na	na	Y	Y	Y	Y	na	na	na	na	Y	Y	na	na	na	Y	na	na
XFI	na	na	na	na	na	na	na	na	na	na	na	na	Y	na	na	Y	Y	na	na	na	Y	na	na

Y – Feature is supported
Blank – Feature is not supported
na – Feature is not available

freescale™

OOBE Issues & Why Virtual Environment

#FTF2015

Linux SDK Looks Easy to Install and Use

1. Mount the ISO on your machine:

```
$ sudo mount -o loop QorIQ-SDK-<version>-<target>-<yyyymmdd>-yocto.iso /mnt/cdrom
```

2. As a non-root user, install Yocto Project:

```
$ /mnt/cdrom/install
```

3. When prompted to input the install path, ensure that the current user has the correct permission for the install path.

There is no uninstall script. To uninstall Yocto Project, you can remove the directory manually.

```
<yocto_install_path>/QorIQ-SDK-<version>-<yyyymmdd>-yocto
```

However, You Have to Know…

- Yocto Project supports typical Linux distributions: Ubuntu, Fedora, CentOS, Debian, OpenSUSE, etc. More Linux distributions are continually being verified.

- This SDK has been verified on following Linux distributions
  - Ubuntu 14.04,
  - Centos-7.0.1406,
  - Mint-15,
  - Debian 7.6,
  - Fedora 20
  - OpenSUSE 13.2

#FTF2015

Therefore, You Have to Prepare the Host Environment - I

Host Environment

Yocto Project requires some packages to be installed on host.

The following three steps are used to prepare the Yocto Project environment.

1. `$ cd <yocto_install_path>`

2. `$ ./scripts/host-prepare.sh`

The script "host-prepare.sh" is used to install the required packages on your host machine. Root permission and Internet access are required to run the script. The script only needs to be run once on each host.

In general, Yocto Project can work on most recent Linux distributions with Python-2.7.3 or later and required packages installed. The default Python is not 2.7.x on some Linux distros, e.g. CentOS 6.5 installs python 2.6.6. Please follow below instructions to install the Python 2.7.x in custom path instead of override the system default python, the override may cause system utilities breaking.

```
$ wget https://www.python.org/ftp/python/2.7.6/Python-2.7.6.tar.xz
[NOTE: Python 2.7.3 and python 2.7.5 can be used as well.]
$ tar -xf Python-2.7.6.tar.xz
$ cd Python-2.7.6
$ ./configure --prefix=/opt/python-2.7.6
$ make
$ sudo make install

Please run below export command to ensure python 2.7.x is used for Yocto build.
$ export PATH=/opt/python-2.7.6/bin:$PATH
```

Yocto Project supports typical Linux distributions:Ubuntu, Fedora, CentOS, Debian, OpenSUSE, etc. More Linux distributions are continually being verified. This SDK has been verified on following Linux distributions: Ubuntu 14.04, centos-7.0.1406,Mint-15,Debian 7.6, Fedora 20 and OpenSUSE 13.1

For a list of the Linux distributions tested by the Yocto Project community see SANITY_TESTED_DISTROS in poky/meta-yocto/conf/distro/poky.conf.

The following is the detailed package list on the Redhat and Centos hosts:

```
$ sudo yum groupinstall "Development Tools"
$ sudo yum install tetex gawk sqlite-devel vim-common redhat-lsb xz
python-devel  zlib-devel perl-String-CRC32 dos2unix python m4 make wget curl ftp
tar bzip2 gzip unzip perl texinfo texi2html diffstat openjade docbook-style-dsssl sed docbook-style-xsl
docbook-dtds docbook-utils bc glibc-devel pcre pcre-devel groff linuxdoc-tools
patch cmake tcl-devel gettext ncurses apr SDL-devel  libtool xterm
mesa-libGL-devel mesa-libGLU-devel gnome-doc-utils autoconf automake
```

Prepare the Host Environment - II

For the Fedora hosts:

```
$ sudo yum groupinstall "Development Tools"
$ sudo yum install tetex gawk sqlite-devel vim-common redhat-lsb xz
python-devel  zlib-devel perl-String-CRC32 dos2unix python m4 make wget curl ftp
tar bzip2 gzip unzip perl texinfo texi2html diffstat openjade
docbook-style-dsssl sed docbook-style-xsl docbook-dtds docbook-utils
bc glibc-devel pcre pcre-devel groff linuxdoc-tools patch cmake
tcl-devel gettext ncurses apr SDL-devel mesa-libGL-devel  xterm
mesa-libGLU-devel gnome-doc-utils autoconf automake libtool
$ sudo yum install ccache quilt perl-ExtUtils-MakeMaker ncurses-devel
```

For Ubuntu and Debian hosts:

```
$ sudo dpkg-reconfigure --terse -f readline dash
$ sudo apt-get install sed wget subversion git-core coreutils unzip
texi2html texinfo libsdl1.2-dev docbook-utils fop gawk python-pysqlite2
diffstat make gcc build-essential xsltproc g++ desktop-file-utils
chrpath libgl1-mesa-dev libglu1-mesa-dev autoconf automake groff libtool
xterm libxml-parser-perl vim-common xz-utils cvs tofrodos
libstring-crc32-perl ubuntu-minimal ubuntu-standard patch
libbonobo2-common libncurses5-dev
```

Extra packages are needed for Ubuntu-64b:

```
$ sudo apt-get install ia32-libs lib32ncurses5-dev
```

For OpenSUSE host:

```
$ sudo zypper install python gcc gcc-c++ libtool
subversion git chrpath automake make wget xz
diffstat texinfo freeglut-devel libSDL-devel patch
```

```
$ source ./fsl-setup-poky -m <machine>
```

For example:
```
  $ source ./fsl-setup-poky -m t2080rdb
```

#FTF2015

Prepare the Host Environment - III

The following shows the usage text for the `fsl-setup-poky` command:

Usage:

```
source fsl-setup-poky <-m machine> [-l] [-h]
```

Where:

    `<-m machine>` is mandatory; `[-j jobs]`, `[-t tasks]`, `[-s string]` ,`[-p]`, `[-l]`and `[-h]` are optional

    Supported QorIQ (PowerPC) machines: t1040qds-64b t1040qds b4420qds-64b b4420qds b4860qds-64b b4860qds bsc9131rdb bsc9132qds c293pcie p1010rdb p1020rdb p1021rdb p1022ds p1023rdb p1025twr p2020ds p2020rdb p2041rdb p3041ds p4080ds p5020ds-64b p5020ds p5040ds-64b p5040ds t1024qds-64b t1024qds t1024rdb-64b t1024rdb t1040rdb-64b t1040rdb t1042rdb-64b t1042rdb t1042rdb-pi-64b t1042rdb-pi t2080qds-64b t2080qds t2080rdb-64b t2080rdb t4160qds-64b t4160qds t4240qds-64b t4240qds t4240rdb-64b t4240rdb

    Supported Layerscape (ARM) machines: LS1021AQDS and TWR-LS1021A

    `[-j jobs]`: number of jobs for make to spawn during the compilation stage.
    `[-t tasks]`: number of BitBake tasks that can be issued in parallel.
    `[-d path]`: non-default path of DL_DIR (downloaded source)
    `[-c path]`: non-default path of SSTATE_DIR (shared state Cache)
    `[-b path]`: non-default path of project folder (build_${machine}_release)
    `[-l]`: lite mode. To help conserve disk space, deletes the building directory once the package is built.
    `[-s string]`: append an extra string to project folder.
    `[-p]`: append fsl cache and source mirrors (For FSL Internal Use Only)
    `[-h]`: help

Previous topic: Install the SDK
Next topic: Builds

Other Software - QorIQ Configuration and Validation Suite
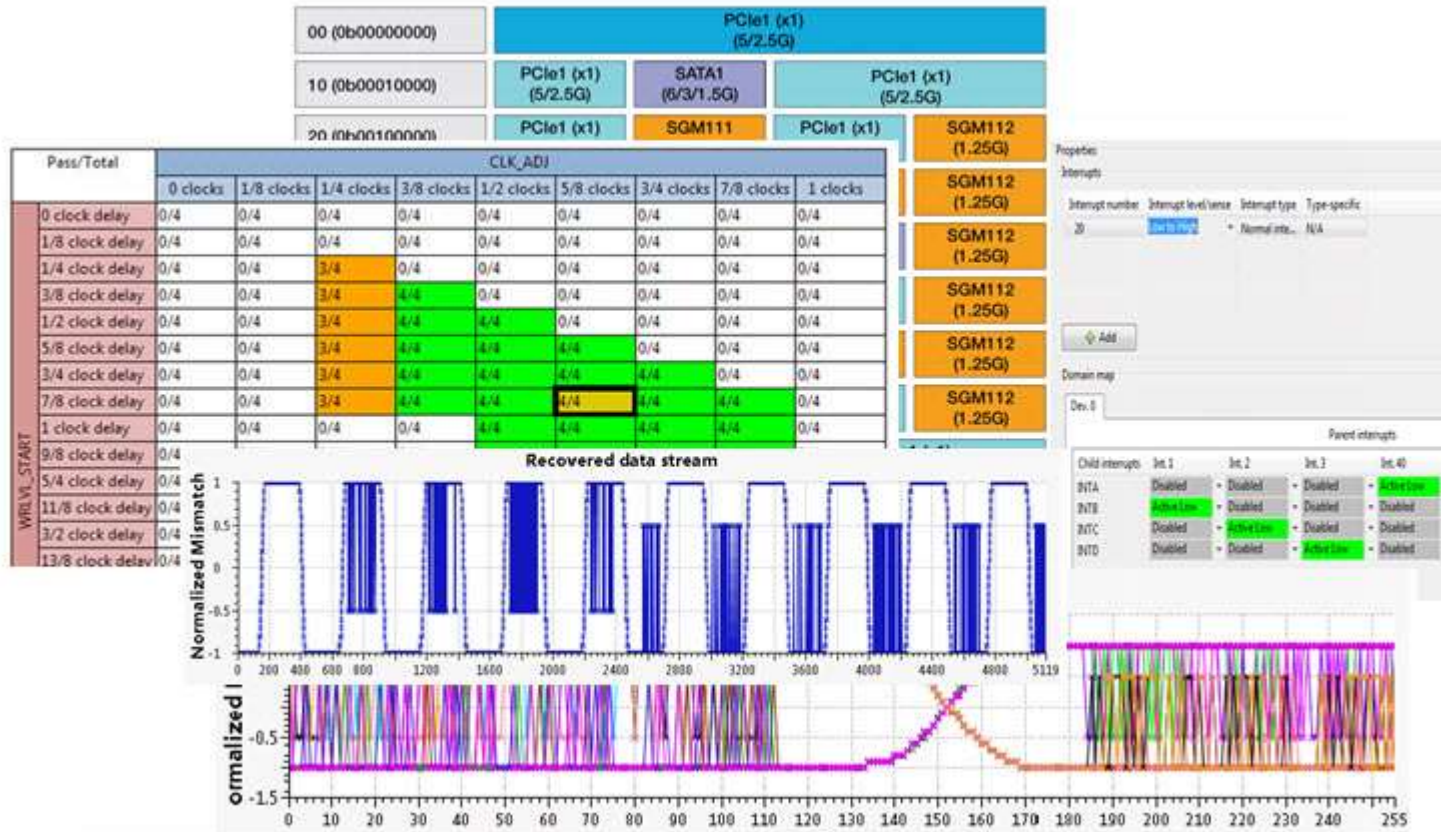
Other Software - QorIQ Configuration and Validation Suite

- QorIQ Configuration and Validation Suite targets engineers doing board design and board bring-up with designs using Freescale QorIQ System-on-Chip (SoC) processors, including the brand new LS series devices. It provides tools that assist in the configuration and validation of key hardware and software SoC features. It is a GUI-based product built on top of Eclipse (www.eclipse.org) and deployed as an Eclipse feature.

- QorIQ Configuration and Validation Suite (QCVS) consists of two separate software packages:
  - The QorIQ Configuration Suite (QCS)
  - DDR Validation Tool

Other Software – CodeWarrior Development Studio PA v10

CodeWarrior Development Studio v10 for Power Architecture® Technology (Eclipse)

#FTF2015
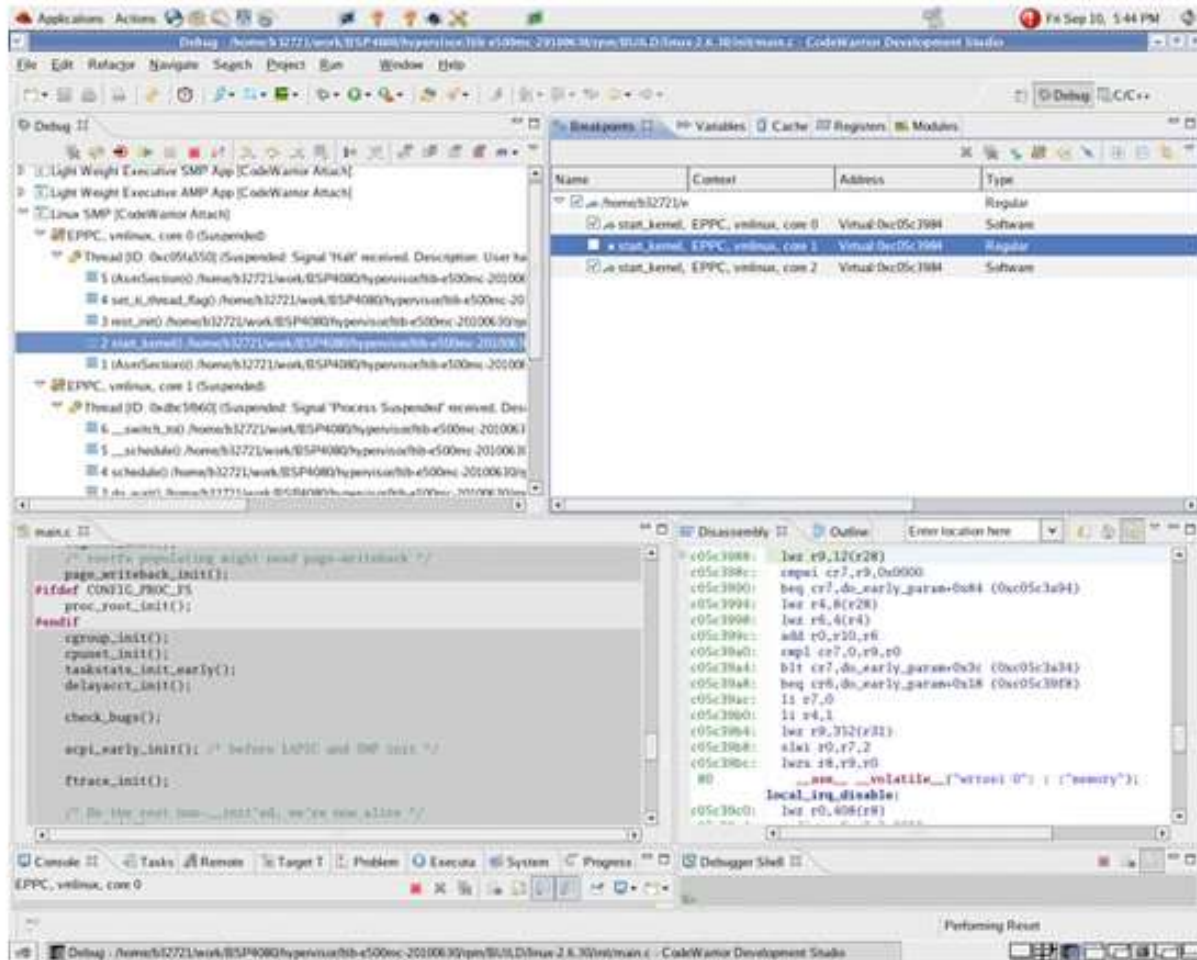
Other Software – CodeWarrior Development Studio PA v10

Overview

Reach the full potential of your communications application with CodeWarrior v10.0 for Power Architecture® development tools for the Freescale QorIQ processors built on Power Architecture® technology. Integrated within an Eclipse framework, the CodeWarrior Development Studio for Power Architecture technology combines GNU build tools and highly advanced asymmetric multiprocessor (AMP)/symmetric multiprocessor (SMP) Linux® multicore debugging with software analysis capabilities, allowing you to build, debug and maximize the performance of Power Architecture-based multicore applications.

Highlights

- Feature-rich, Eclipse-based IDE
- Multicore AMP/Linux SMP debugger with multicore run-control commands
- Highly advanced software analysis tools, including program and data path acceleration architecture (DPAA) trace tools and extended, integrated support for popular open source tools such as the GNU Linux Trace Tool (LTTng), OProfile and Valgrind

- Support for Linux application and kernel development (tool capabilities vary with the targeted processor)

Eclipse IDE

- Common IDE platform
- Open standard
  - Common framework
  - Leverages larger ecosystem
- Extensible
  - Eclipse-based tools enable extensions via plug-ins
- Ease of use
  - Common look and feel across Freescale Eclipse-based CodeWarrior platforms
  - Eclipse perspectives
- Customizable window layout

Software Analysis[i]

Several of Freescale's QorIQ processors include advanced features such as packet processing accelerators (i.e., parts of DPAA on the P4080). The CodeWarrior Development Studio for Power Architecture technology provides software analysis capabilities that give developers the tools they need to tune application performance or debug complex timing issues issues.

Trace Analysis

For applicable target devices, CodeWarrior trace tools provide developers with timing information for:

- Program flow trace
- Data access
- Packet processing events (available for some processing units)
- Linux kernel events using the LTTng

[1] The software analysis capabilities vary by the processor that is being targeted. This is partially the result of the fact that different processors have different capabilities, including different accelerators and different debug, trace and event counting capabilities.

#FTF2015

Other Software – CodeWarrior Development Studio PA v10

Performance Analysis

Performance analysis tools provide another way to non-intrusively debug functional and performance application issues.

Tools include:

- Configuration of event counters
- Pre-defined "metrics" that provide meaningful performance information for networking applications. Examples include:
  - Cache hit/miss ratios
  - Branch hit/miss ratios
  - Stall cycles per address collision
  - Instructions per cycle
- Various views for event data, including:
  - Average values
  - Timelines
  - Raw data

CodeWarrior Debugger

CodeWarrior multicore debugging support allows the developer to issue commands across user-defined subsets of cores. Special support for Linux kernel and application development is also included.

Multicore debugging features include:

- Multicore run, start and stop commands
- Multicore reset
- SMP debugging
- AMP debugging
- Support for debugging applications using Freescale hypervisor or Light Weight Executive

Build Tools

The CodeWarrior Development Studio v10.0 for Power Architecture technology includes GNU build tools for all e500mc-based cores. For e500v2-based processors, the Freescale CodeWarrior build tools generate optimized code for C.

Documentation

- Linux SDK
- Silicon
- Board
- QCVS
- CodeWarrior

#FTF2015

Linux SDK Bundling in a Virtual Environment

- **Problem**
  - Customer evaluation of processors can be time consuming
  - Installation/Configuration issues

- **Proposal**
  - Provide a configured "machine" – Virtual Environment which limits the steps in building an SDK & other software for early evaluation

- **Benefits**
  - Improves OOBE with Freescale silicon & software
  - Allows Windows-based users to evaluate
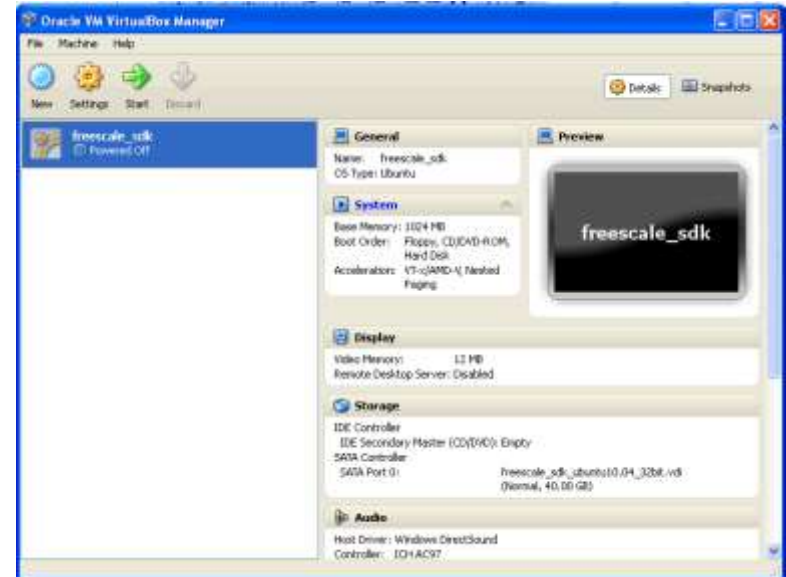
VirtualBox

#FTF2015

Linux SDK Virtual Machine Environment

✓ **KEY CONTENT**

✓ **DEPLOYMENT PROCESS**

✓ **USAGE AND APPLICATION**

✓ **RELEASE AND DISTRIBUTION**

#FTF2015

Linux SDK Bundling in a Virtual Environment

- **Problem**
  - Customer evaluation of processors can be time consuming
  - Installation/Configuration issues

- **Proposal**
  - Provide a configured "machine" – Virtual Environment which limits the steps in building an SDK & other software for early evaluation

- **Benefits**
  - Improve OOBE with Freescale silicon & software
  - Allow Windows-based users to evaluate

VirtualBox

#FTF2015
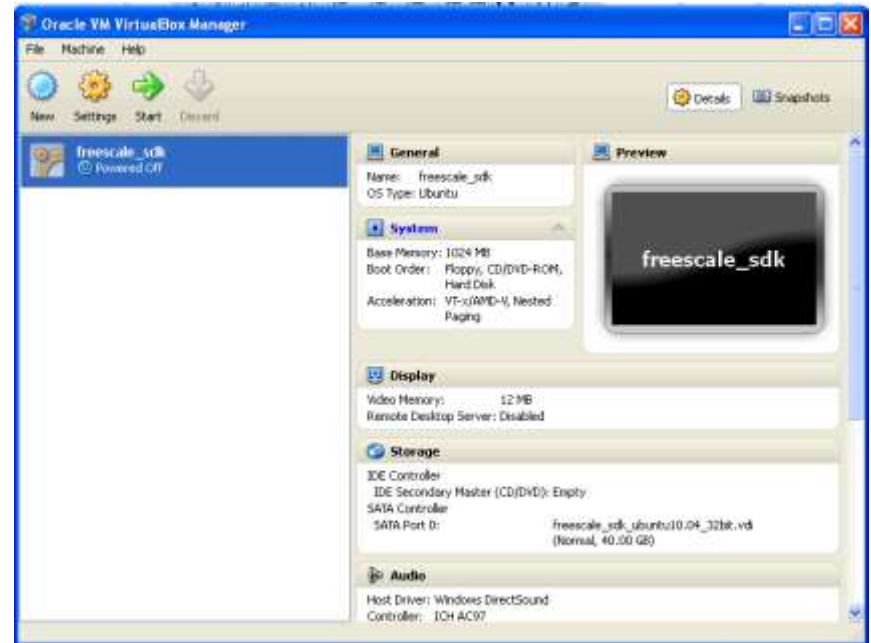
Virtual Machine Overview

In general, Virtual Machines possess four key characteristics that benefit the user:

- **Compatibility:** Virtual Machines are compatible with all standard x86 computers.

- **Isolation:** Virtual Machines are isolated from each other as if physically separated.

- **Encapsulation:** Virtual Machines encapsulate a complete computing environment.

- **Hardware independence:** Virtual Machines run independently of underlying hardware.

#FTF2015

Linux Virtual Machine Environment Content

- The Linux Virtual Machine is built with the VirtualBox and Ubuntu distributions, and contains the entire Freescale Linux SDK development environment:

  - Installed QorIQ Linux SDK
  - Prebuilt target images
  - QorIQ Configuration and Validation Suite
  - Documentation – Silicon RM, Board UM, App Notes, SDK UM, etc.
  - CodeWarrior PA installer and SP , CodeWarrior for ARMv7 Installer.
  - Reference Software
  - Tools

Linux Virtual Machine Environment V1.7 Content

- Ubuntu 14.04 32-bit
- Installed SDK 1.7
- QorIQ Configuration and Validation Suite 4.1.1
- Documentation: Document in the SDK 1.7
- Other Documentation:
  - QorIQ Configuration Suite
  - Publicly released Silicon RM (based on the different platforms)
  - Board RM (based on the different platforms)
- VM RM and release note
- CodeWarrior PA 10.4 installer with SP1
- CodeWarrior ARMv7 10.0.4 installer
- Prebuilt target images
- Reference Software – included within the Linux SDK1.7

#FTF2015

Linux Virtual Machine Images

- The Freescale Linux Virtual Machine is a Virtual Disk Image (VDI) created by VirtualBox containing the Ubuntu operating system and the Yocto utility. You get an entire Linux SDK development environment for Freescale development boards.

- The VDI is named `freescale_sdk_ubuntu14.04_32bit_xxxxxx_yyyyyyyy.vdi`, and you can obtain the zipped file named `freescale_sdk_ubuntu14.04_32bit_xxxxxx_yyyyyyyy.zip`.
  - `xxxxxx`: specific Power Architecture or ARM platform
  - `yyyyyyyy`: creation date of the Linux Virtual Machine

#FTF2015

VE1.7 Images

File	Platform
freescale_sdk_ubuntu14.04_32bit_CORTEXA7_20150108.zip	CORTEXA7 platform
freescale_sdk_ubuntu14.04_32bit_PPC64E5500_20150108.zip	E5500 64bit platform
freescale_sdk_ubuntu14.04_32bit_PPC64E6500_20150108.zip	E6500 64bit platform
freescale_sdk_ubuntu14.04_32bit_PPCE5500_20150108.zip	E5500 32bit platform
freescale_sdk_ubuntu14.04_32bit_PPCE500MC_20150108.zip	E500MC platform
freescale_sdk_ubuntu14.04_32bit_PPCE500V2_20150108.zip	E500V2 platform
freescale_sdk_ubuntu14.04_32bit_PPCE6500_20150108.zip	E6500 platform

- The above seven files represent seven Virtual Machines which are independent to each other.
- Every Virtual Machine is a complete build of the specific platform
- You can use any of the seven Virtual Machines directly according to your needs.
- Example, if you want to use the SDK Virtual Machine for P4080, P3041, P2041, you just need to install the freescale_sdk_ubuntu14.04_32bit_PPCE500MC_20150108.zip and use it directly.

How to Work on Multiple platforms in one Physical Machine

- For example, E500V2 Virtual Machine and E500MC Virtual Machine in one computer.

- You do not need to install multiple Virtual Machines in the computer. That would take up too much host disk space.

- A better approach is to get the new SDK ISO (E500MC) for the specific Power Architecture or ARM platform from Freescale, and to install the new SDK into your original E500V2 Virtual Machine. So your Virtual Machine can support multiple platforms.

Detailed Steps

1. You must install the new SDK into the same directory with the installed SDK in your Virtual Machine:

   `./SDK`

2. Because the SOURCE ISO of the SDK is common for all the platforms and has been installed into your Virtual Machine, you don't need to install again for the new platform

3. For the images in the new SDK ISO, you can make a new directory under:

   `./SDK/Target_Images/`

4. You can find the toolchain file in the SDK ISO, if needed, you can install it to:

   `/opt/fsl-networking`

#FTF2015

Installation and Deployment

1. Get the
freescale_sdk_ubuntu14.04_32bit_xxxxxx_yyyyyyyy.vdi by
unzip **freescale_sdk_ubuntu14.04_32bit_xxxxxx_yyyyyyyy.zip**

#FTF2015
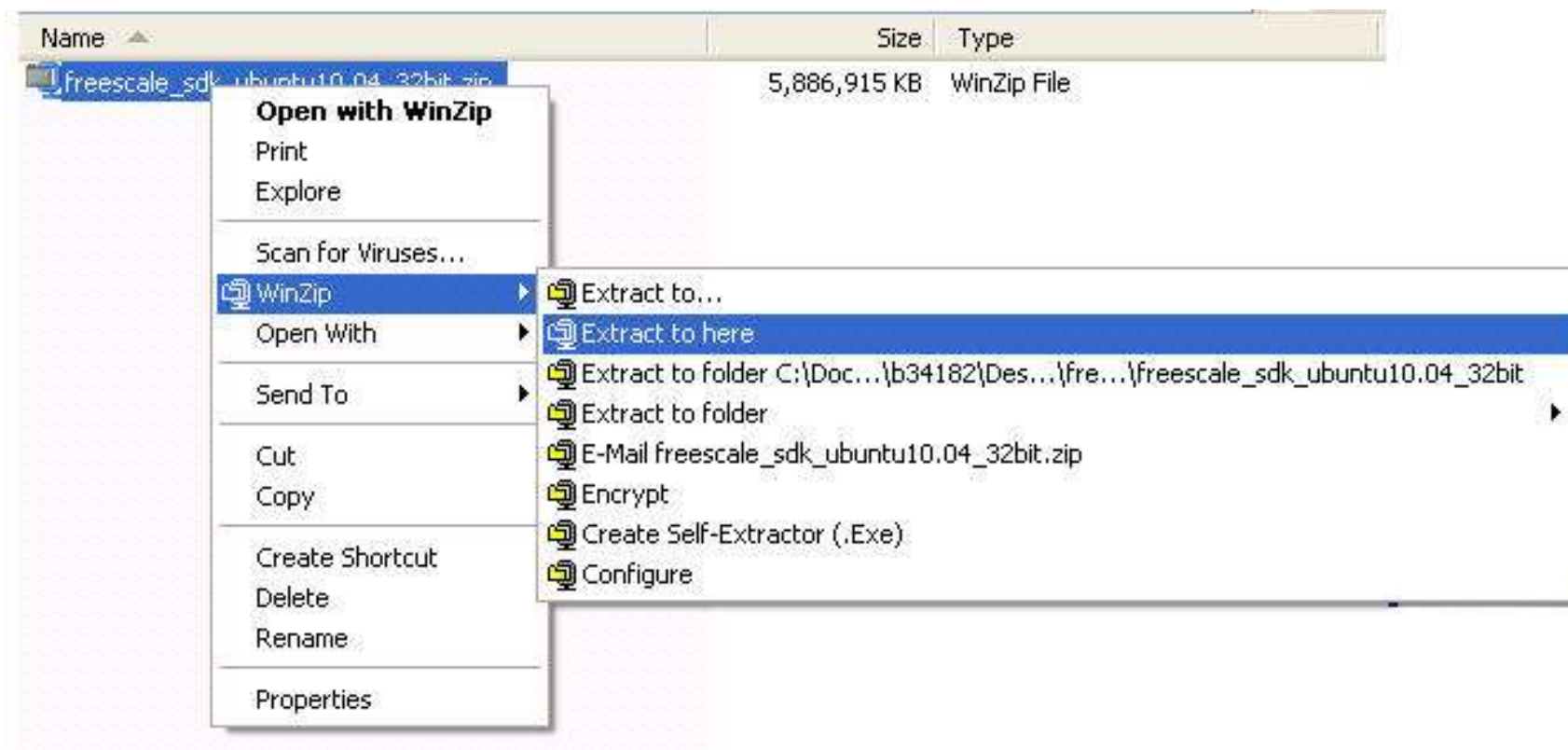
Installation and Deployment

2. Run the VirtualBox installed in your host system.
    Click **New** to get the New Virtual Machine Wizard box

#FTF2015
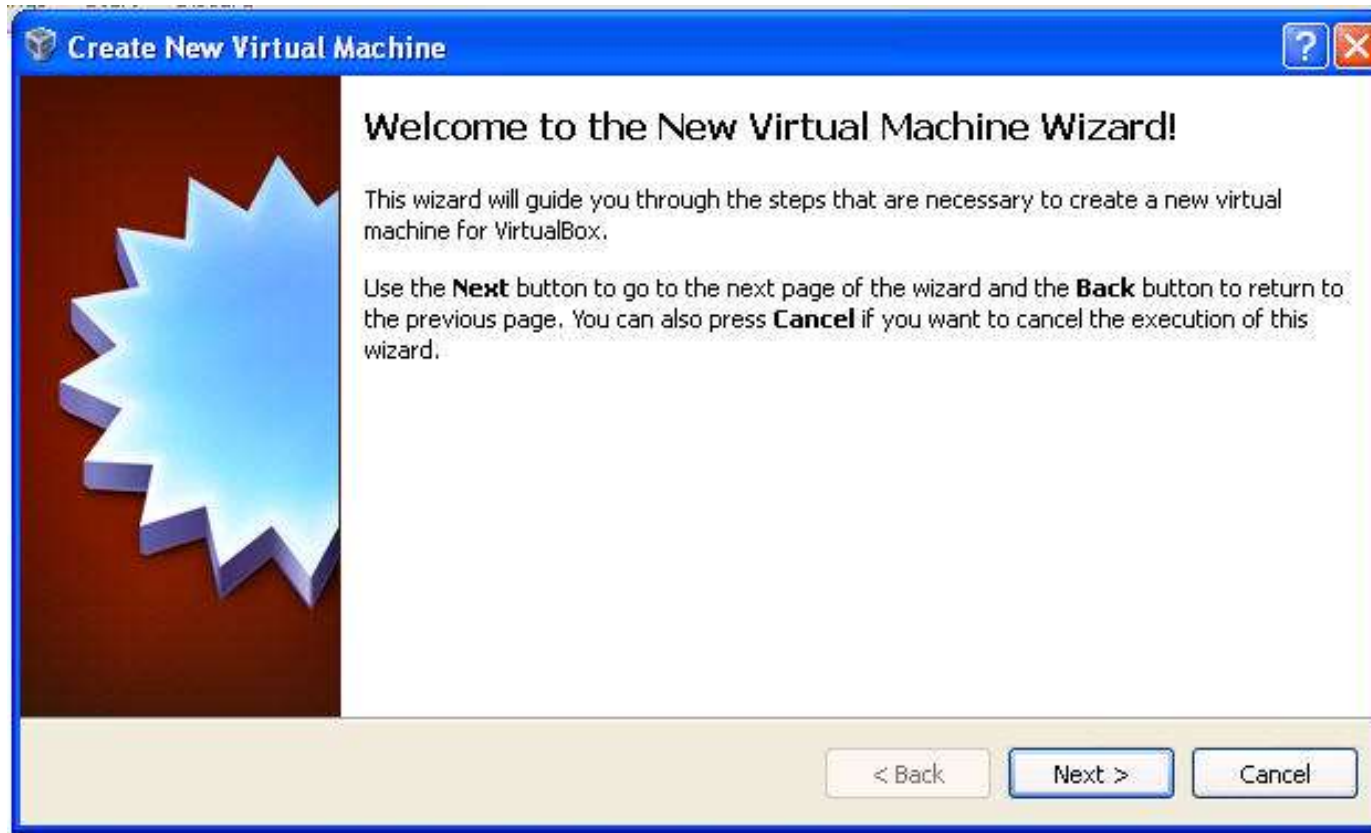
Installation and Deployment

3. Click **Next**. Enter the Virtual Machine's **Name** and select the **Linux Operating System** and Ubuntu (32-bit) **Version**.

#FTF2015
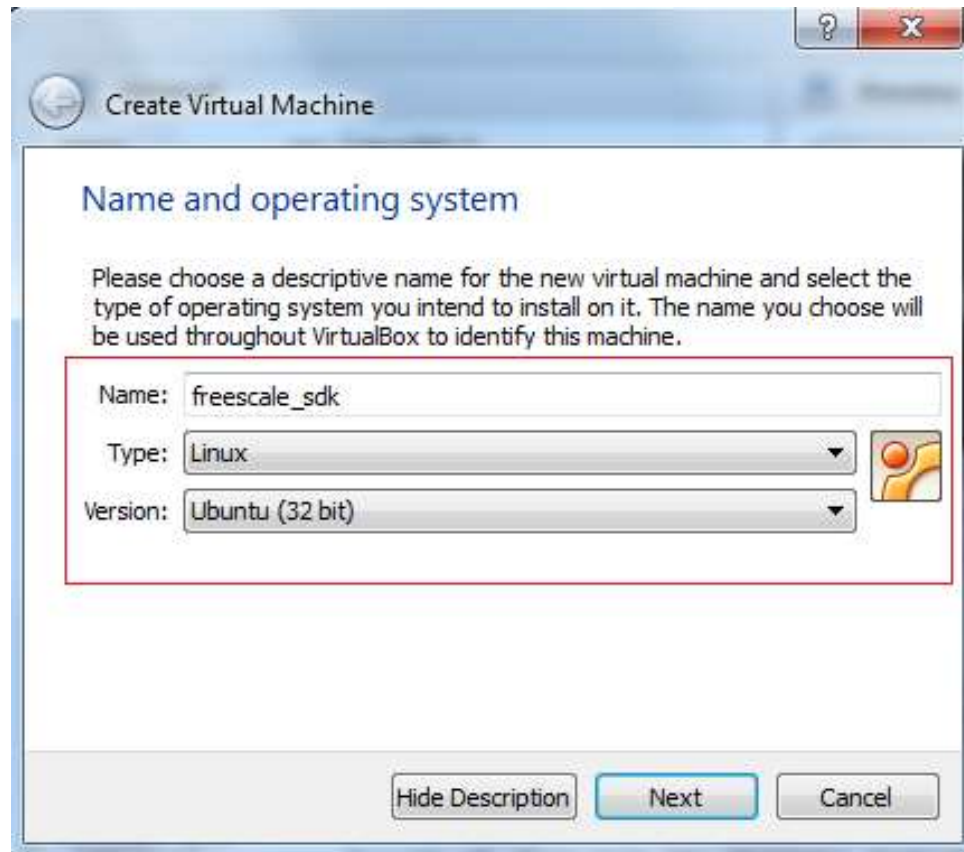
Installation and Deployment

4. Click **Next**. Enter the **Base Memory Size** as 1024MB.
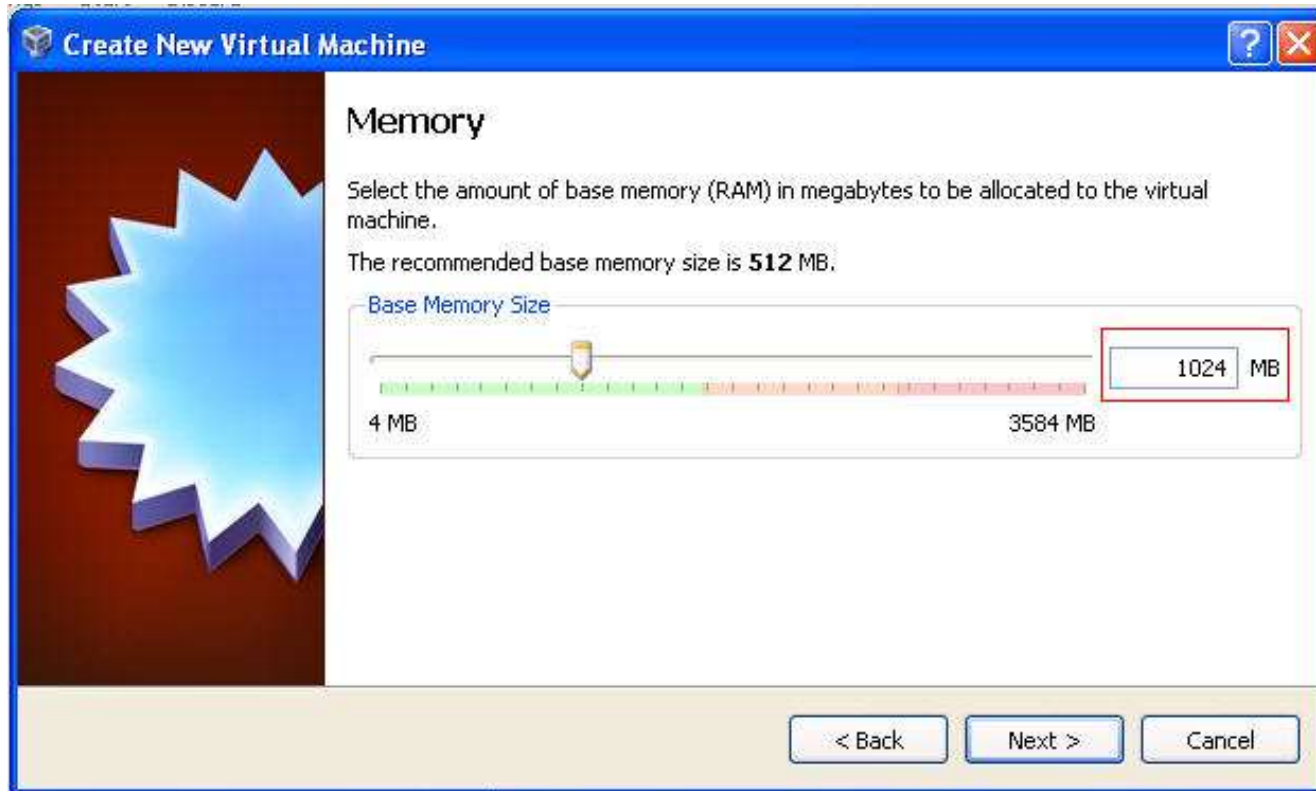
Installation and Deployment

5. Click **Next**. Configure the **Virtual Hard Disk**. Click **Using existing hard disk** with
freescale_sdk_ubuntu14.04_32bit_xxxxxx_yyyyyyyy.vdi file
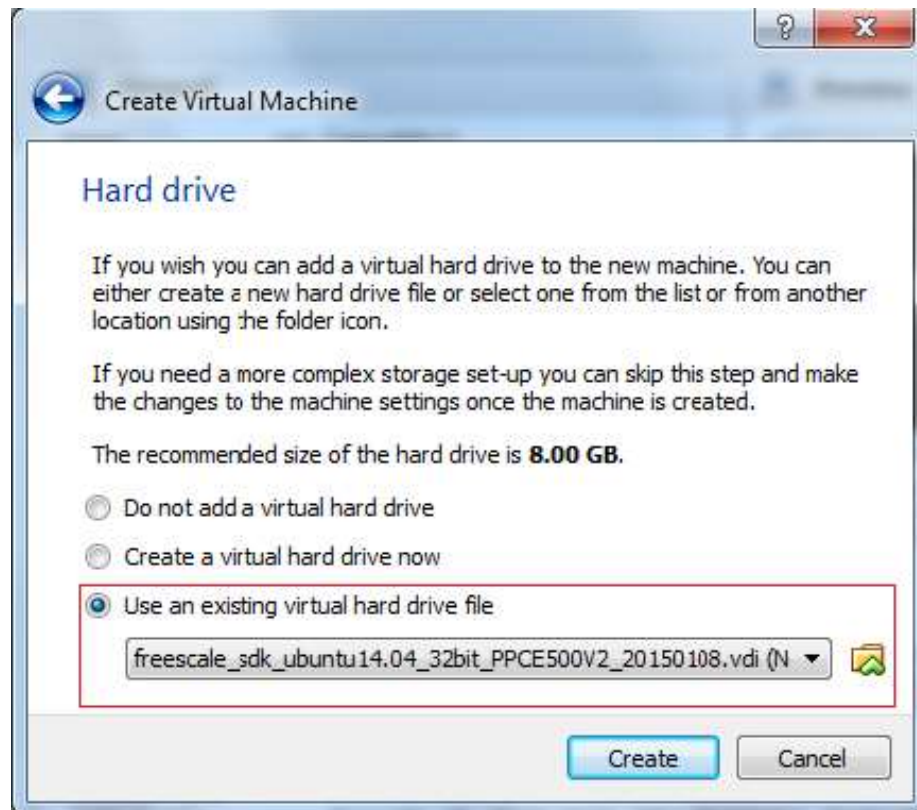
Installation and Deployment

6. Click **Next** to get Summary
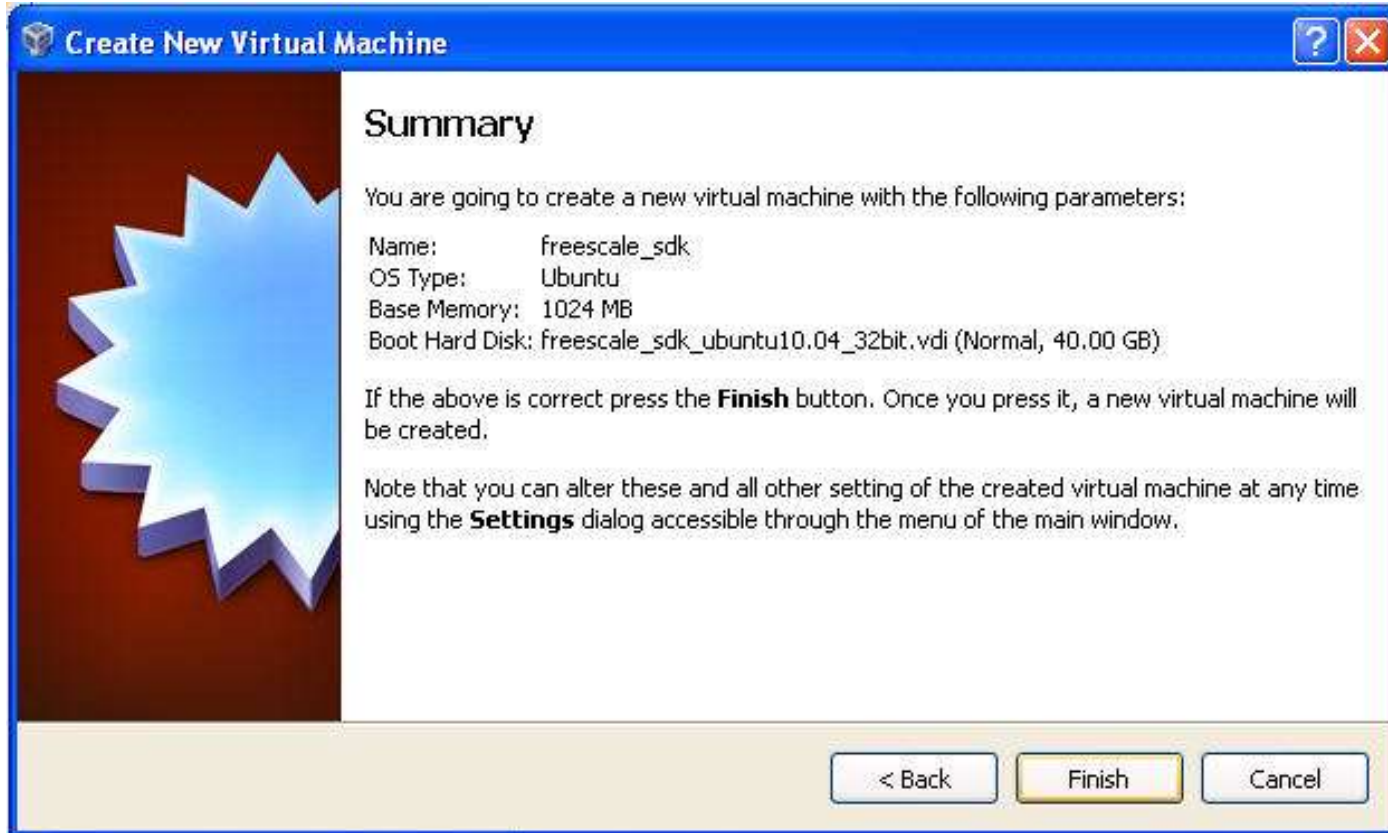
Installation and Deployment

7. Click **Finish** to get the new Linux Virtual Machine

#FTF2015
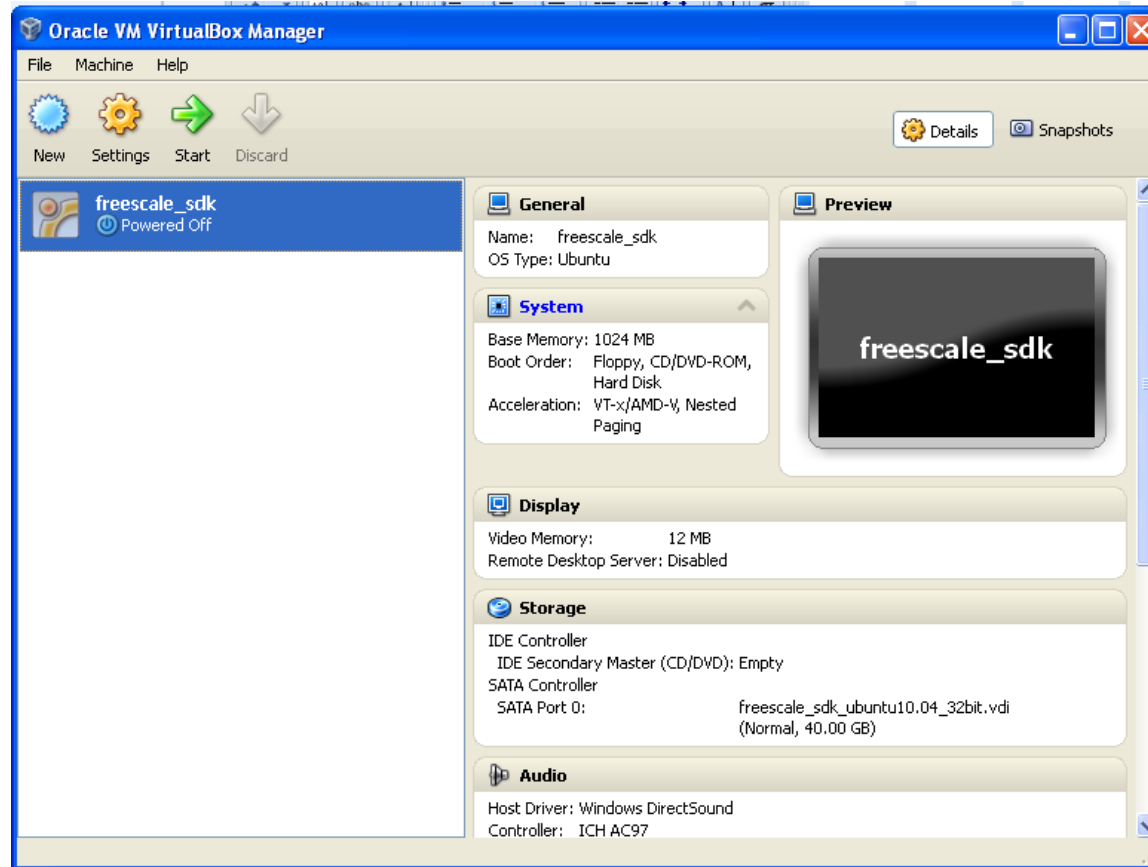
Login

User and password:

Account	Password
root	Freescale
freescale	Freescale

NOTE: The password is case-sensitive.

Directory of User Freescale
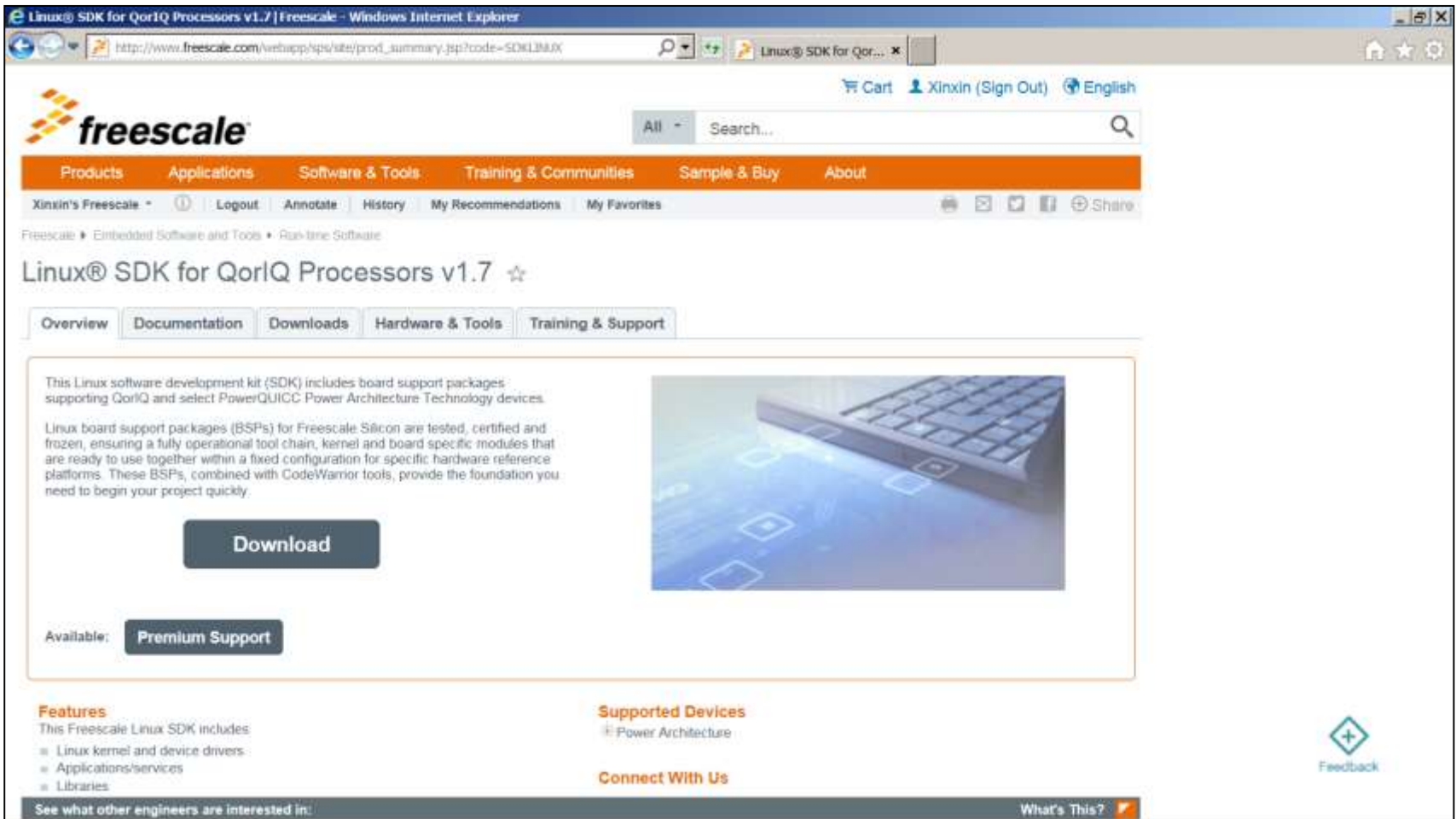
Directory	Description
Home directory of user `freescale`	
`./Documents`	Documents about SDK, silicon, boards, and QorIQ configuration suite.
`./Documents/SDK`	Documents from the SDK ISO. You can get some helpful information when work under the SDK environment.
`./Documents/Silicon_Boards`	Silicon and board reference manuals.
`./Documents/QCVS`	Documents about QorIQ Configuration and Validation Suite.
`./Fsl_tools`	This directory includes Freescale Linux development tools. Now includes a CodeWarrior installer.
`./SDK`	This directory includes an installed Yocot environment for QorIQ SDK.
`./SDK/Target_Images`	For the specific PowerPC or ARM platform, this directory includes all images which can be used directly on the specific PowerPC or ARM development boards. For example, there are flash images, uboot images, DTBs, RCWs, rootfs images.

Directory of User Freescale

Directory	Description
Other useful directories for Linux BSP development	
`/opt/Freescale/`	Installed QorIQ Configuration and Validation Suite based on Eclipse. It contains some configuration tools for QorIQ architecture such as PBL. This suite was built based on Eclipse IDE for C/C++ Developers and the QorIQ Configuration suite package. Using this Configuration Suite, you can run "eclipse" directly.
`/opt/fsl-networking`	Installed tool-chain for the specific PowerPC or ARM platform.

How to Get the VE Files

Visit: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=SDKLINUX

#FTF2015

How to Get the VE Files

#FTF2015

Downloaded SDK and VE in 2014

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
SDK1.5 / SDK1.6	255	229	303	314	292	438	471	307	371	259
VE1.5 / VE1.6	139	118	153	152	187	109	179	158	157	146
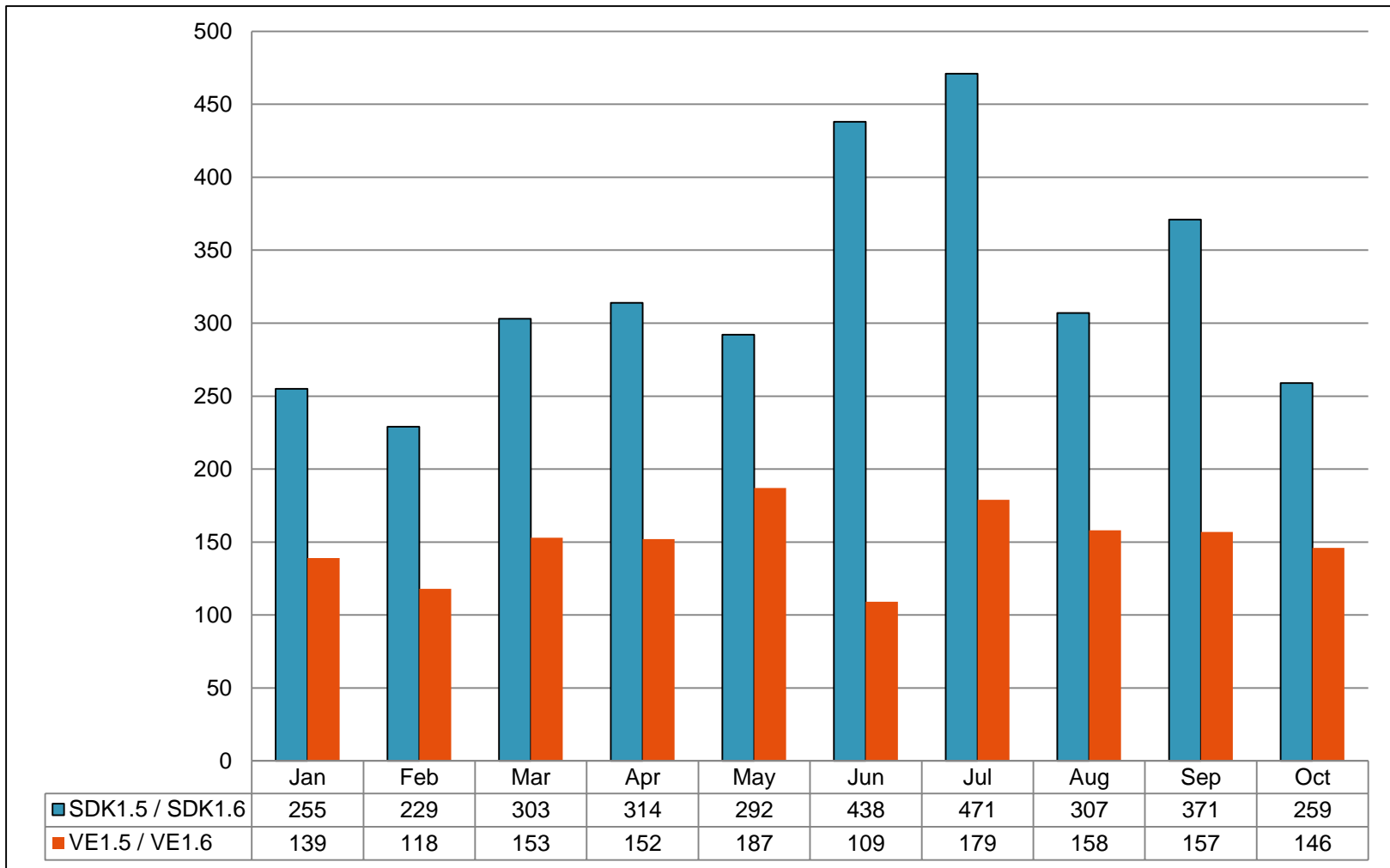
freescale™

Customer's Feedback

- Makes it easy and simple to set up the environment/configuration to start the software exploration and reproduce what the Freescale engineering team intended

- One place to find all software and related documents

- File size itself too large to be fast downloaded.

- Some customer IT departments expressed concern regarding the guest OS running

Summary

- Introduction to Freescale QorIQ Linux SDK
- OOBE Issues and why Virtual Environment
- Linux SDK Virtual Machine Environment
  - Key Content
  - Deployment Process
  - Usage and Application
  - Release and Distribution
- Summary
- Q&A

#FTF2015

www.Freescale.com