Release Notes
For
Switch Software Development Kit

SDK 6.5.6

TABLE OF CONTENTS

Section 1: About This Document

These are the Release Notes for the Broadcom Network Switching Software Development Kit Release 6.5.6.

This document provides a general description of the release and its new features. It also describes the chips supported by the release, BCM API additions or changes, resolved issues, and any relevant open issues. The reader should refer to prior release notes for 6.5.x, as only new features or issues are described in this version of the release notes.

Section 2: Product Documentation

The following documents are available through Broadcom's Customer Support Portal, http://support.broadcom.com. They are the primary source of information and should be referenced when using this release:

Document	***Description***
56XX-PG656-R	*BCM API Reference Guide* This manual describes the theory of operations of the API and all existing BCM APIs for this release.
56XX-PG820-R	*Network Switching Software Platform Guide* This guide describes the SDK source and Makefile structure, abstraction and porting layers, device specific interactions, and the platform/operating system specific features of the SDK. If this is your first time working with the SDK, start with this document.
56XX-PG707-R	*Stacking Guide* This guide describes how to use the discovery and stacking applications provided in this release.

Section 3: New Devices added to this release

For any given SDK release, support for certain devices may be provided in Preview or Supported status. Devices in “Supported Switch Devices” and “Supported PHYs” have completed the full QA process and are intended for use in production systems. It is expected that customers would integrate the version of the SDK which provides "Supported" status for their use on actual development or production systems.

Devices in “Preview Switch Devices” and “Preview PHYs” are provided to allow early integration of the customer's application with the SDK APIs that support that device. This software has not been fully tested on the physical target device and should not be expected to fully function.

Section 3.1: Supported Switch Devices

Family	*Devices*	*Description*
BCM56760	BCM56760 B0	6x40GbE+ 48x10GbE Ethernet Embedded Switch
	BCM56762 B0	24x10GE + 6x40GE with support for 25GE
	BCM56765 B0	24x10GE + 6x40GE with support for 25GE
	BCM56768 B0	48x10GE + 4x100GE

Section 3.2: Preview Switch Devices

Family	*Devices*	*Description*
BCM56970	BCM56970 A0	6.4 Tbps Multilayer Switch
BCM88060	BCM88060 A0	Fibre Channel/Fibre Channel-over-Ethernet Mapper
	BCM88061 A0	Fibre Channel/Fibre Channel-over-Ethernet Mapper
BCM88470	BCM88470 B0	32x12.5GE + 16x25GE Ethernet Switch
BCM88270	BCM88270 A0	28x2.5GE + 8x10.3GE Ethernet Switch

Section 3.3: Supported PHYs

Family	*Devices*	*Description*	*Switch qualified against*

BCM82332	BCM82332	1x 100G/3x 40G/12x 10G Ethernet PHY	BCM56860
	BCM82793	100G Gearbox for 100GBASE CR4/SR4/LR4 & SR10	BCM56860

Section 3.4: Preview PHYs

Family	*Devices*	*Description*	*Switch qualified against*
BCM84848	BCM84888 A0	10G/5G/2.5G/1000/ 100BASE-T Transceiver	BCM56860
BCM54190	BCM54190 A0	Octal SGMII Copper Gigabit Ethernet Transceiver	

Section 4: New Features per Device

Section 4.1: BCM56760 (MAVERICK) B0 FAMILY GENERAL AVAILABILITY (GA) SUPPORT

The BCM56760 family is a class of high performance, nonblocking network switching devices supporting up to 720Gbps to address performance, capacity and service requirements of next generation data center and cloud computing environment. They feature multiple integrated TSC Serdes core with support for 100GE delivering high-bandwidth, glueless network connectivity. This release supports the following port configurations:

For BCM56760:

48x10GbE + 6x40GbE (ToR configuration)
72x10GbE
4x50GbE + 48x10GbE + 1x10GbE
2x100GbE + 4x40GbE+ 8x25GbE + 12x10GbE + 2x10GbE

For BCM56765:

72x10GbE
2x100GbE + 48x10GbE
16x25GbE + 8x40GbE
48x10GbE + 6x40GbE

18x40GbE

This release provides GA support for BCM56760 B0, BCM56762 B0, BCM56765 B0, and BCM56768 B0 devices.

Section 4.1.1: Additional Features Supported

Section 4.1.1.1: MMU threshold settings for CT mode

In this release, PFC optimized MMU settings to support Cut-Through mode have been updated.

Section 4.2: BCM56970 (Tomahawk2) Preview support

The Broadcom® BCM56970 family is a class of high performance, high-connectivity network switching devices supporting up to 64x 100GbE, 128x 40GbE, 128x 25GbE, or 128x 10GbE switch ports. The BCM56970 delivers high-bandwidth, glueless network connectivity up to 6.4 Tbps on a single chip. In this release, preview support is provided for the BCM56970 A0 device.

Section 4.2.1: Preview features supported

The table below shows the status of supported features in SDK 6.5.6. These features have not yet been tested with actual silicon.

Table 1: SDK Feature Status

Feature	***Status***	***Note***
L2	Preview	
L3	Preview	
VLAN	Preview	
Port	Preview	Support 16nm Serdes
Flex Port	Preview	Legacy lane set API deprecated
STG	Preview	
Trunk	Preview	
Mirror	Preview	
Switch Controls	Preview	
Legacy MPLS	Preview	
VxLAN	Preview	
Multiple split horizon group	Preview	
Mac in Mac	Preview	
Proxy	Preview	

Higig Proxy	Preview	
Legacy Stats	Preview	
Flex Counters	Preview	ALPM flex counter support added
Qos Assignment	Preview	
Rate	Preview	
Meters	Preview	
Policer	Preview	
IPMC	Preview	
Multicast	Preview	
Legacy Subports	Preview	
VP-VLAN Membership checks	Preview	
NIV	Preview	
Port Extender	Preview	
Trill	Preview	
NAT	Preview	
VPLAG	Preview	
Legacy Protection Switching	Preview	
Field Processor	Preview	Added new actions and qualifiers
UDF	Preview	
System and Network level congestion	Preview	
Packet Tx/Rx	Preview	
Base MMU	Preview	
COSQ	Preview	Added buffer ID support for MMU threshold and WRED settings
BST	Preview	Added buffer ID support; added 2 new BST objects
RIOT	Preview	
SER	Preview	
Visibility: AGM	Preview	
Visibility: Packet Trace	Preview	
LPM	Preview	
ALPM	Preview	Enhanced ALPM utilization

RTAG7 flex hash	Preview	Support all 13 RTAG7 BINs to extract from UDF
Resilient Hash	Preview	
ECMP and Hierarchical ECMP	Preview	
UFT	Preview	Optimized breadth-first hash reordering algorithm
Time	Not supported yet	
BFD/BHH	Not supported yet	

Section 4.2.2: New and enhanced features supported on BCM56970

The following features have been added to this release as preview only and will be supported with the BCM56970 chip.

Table 2: New feature support

Feature	***Note***
DLB over ECMP, HIGIG, LAG	Provide a dynamic load balancing scheme by considering link state when assigning flows to members of HiGig Trunk, Link Aggregation, and L3 ECMP.
DGM (Dynamic Group Multipath	Provide an adaptive routing solution by incorporating the path quality metrics into L3 ECMP forwarding decision.
TCB (Transient Capture Buffer)	Focus and capture a number of additional transactions associated with the drop entity. This will provide information before and after the trigger event
Packetized statistics	Provide an alternative to the OOB stats to report MMU queue/pool occupancy
Packet timestamping	Provide capability of attaching arrival and departure times to a frame along a flow path across the network
DPP	Decoupled Packet Processor
MPLS Enhancements	● MPLS ECN Marking ● MPLS Label Swap ● MPLS Segment Routing ● MPLS FRR (Fast Reroute) Lookup ● MPLS 5 Labels Parsing
SER TCAM scan time reduction	Expedited TCAM memory scan conducted by software

Section 4.3: BCM56965 (Tomahawk+) family updates

Section 4.3.1: Software Auto Negotiation

Software Auto Negotiation is supported on BCM56965 A0 devices only. When this feature is enabled the device can advertise and negotiate MSA 25G/50G abilities including FEC. If the link partner device advertises at IEEE speeds 10G/40G/100G, the device will be forced to Hardware Auto negotiation mode.

New SOC property “sw_autoneg_polling_interval” has been added. Default value is 50ms. In addition existing SOC property “phy_an_c73” has been enhanced to support values 3 (meaning software auto negotiation must advertise both IEEE and MSA abilities) and 4 (meaning software auto negotiation must advertise MSA abilities only).

Section 4.4: BCM88060 (Montreal2) A0 Beta support

The Broadcom BCM88060 is a second generation Fibre Channel (FC) and Fibre Channel-over-Ethernet (FCoE) mapper. The device enables a flexible chip to be deployed in front of a data center switches to connect to FC endpoints directly. This device provides the flexibility to support FC/FCoE/Ethernet universal port capability as needed. The device incorporates the FC mapping into and out of Ethernet (FCoE) in a physical layer device. As a universal port device, the FCoE capability can optionally connect to a host of switches via a standard Ethernet when the FC universal port is needed.

This release supports the following device firmware: BCM8806x version 1.0.6 (rc/firmware/BCM8806x) which should be copied to the SDK boot directory. This firmware binary (BCM8806x) must be present in the SDK boot directory for loading during initialization.

Section 4.4.1: Status of supported features

Bring up and testing of the following BCM88060 features have been completed in this SDK release. Some features have reached Beta quality as they have already entered our test regression phase. Table 2 shows the status of features in this release.

Table 3: SDK Feature Status

Feature	Status	Note
Ethernet Retimer mode - Forced	Beta	Supported speeds: 10G / 20G / 25G / 40G / 50G / 100G
Ethernet Retimer mode - Autoneg	Beta	Supported speeds: 10G / 20G / 25G / 40G / 50G / 100G
Ethernet Gearbox mode - Autoneg/Forced	Beta	Supported 40G : System side 2x20G , line side 4x10G
Ethernet EBE (Extended Buffered for Ethernet mode)	Beta	Supported speeds: 10G / 20G / 25G / 40G / 50G / 100G Supports different speeds for line and system side, but lane counts should be same.
Warmboot	Beta	Tested in Linux environment with Ethernet and FCoE.
Ethernet Flexport	Beta	Quad, Tri0, Tri1, dual, single port mode supported.
HiGig	Beta	Supported 42G and 106G
Boot optimization	Beta	Tested boot and initialization timings with multiple MT2 setup.
Parallel Download with multiple MT2	Beta	Tested with chained MT2 setup.
FCoE - Fixed mode and AN mode with TTS	Beta	Supported speeds: 4G / 8G / 16G / 32G.
FCoE - Flexport	Beta	System side flex supported.
FCoE - Statistics and Events handling	Beta	Tested inband statistics in FCoE setup.
Dynamic switch of Ethernet and FC modes	Bringup	Supported.
Diagnostics	Beta	Register read/write, dsc dump, eye scan, PRBS supported. Phy diag supports multistage loopback. FW logging supported.

Traffic tests	Beta	TR19, TR72 supported.

Section 4.4.2: Notes and Known Limitations

Tested with BCM956960 & BCM56850 SVK on PowerPC with Linux.
Reset of port required after PRBS tests.
FCoE COS to priority API not supported.

Section 4.5: BCM88470 (Qumran-AX) Family Beta Release

This release is a beta version for the BCM88470-Family product line, following previously released 6.5.5 Beta version.
The subsequent sections describe the features validated for this release, known issues and bring-up guidelines. In addition, 6.5.6 is the first release which introduces Qumran-AX B0 revision. For more information about Qumran-AX B0 revision please see BCM88470-Family B0 revision support. It is extremely important to review “Backward compatible important notes” section before starting the integration of the new release.
Note: IPsec SW package is available as a separate add-on upon request.

Section 4.5.1: Backward compatible important notes

See BCM88470-Family “Backward compatible important notes” section.

Section 4.5.2: Validated Features

Basic data path, connectivity and Traffic Management features:

Register & Memory access including DMA
Supported SKUs:
- Qumran-AX: 88470, 88470P, 88471,88473
- Kalia: 88476, 88476P

Device core mode: dual-core symmetrical
Interfaces:

- SRAM, DDR4@1.6GHz
- CPU RX/TX (packet DMA)
- NIF 156.25MHz and 125MHz
- NIF:
  - 10GE: KR/XFI, RXAUI, XAUI
  - 40GE: XLGE
  - 100GE
  - ILKN

Only 12.5G.
Eagle and Falcon on same port isn't allowed.
- QSGMII

Forwarding:
- Unicast
- Multicast
- Mirroring

CosQ:
- End-to-end scheduling
- Ingress and Egress compensation
- VOQ ingress queue creation. Mapping by-destination and by-flow-id
- Tail Drop
- Credit watch dog
- Egress queuing: number of priorities per port: 1/2/8
- Undersubscribed CoE (mapping multiple ports, no Flow Control)

Counters:
- SNMP counters
- Internal (diagnostics) counters

Packet forwarding - ITMH, Force-FWD
Fabric connectivity with FE device (Kalia only)
Link Bonding

Packet Processing:
- Diagnostics
- Port
- STG
- L2-Forwarding
- L2-Learning
- VLAN-Translation
- VLAN
- VSWITCH
- L3-IPv4
- L3-IPv6
- L3-Interface-RIF
- L3-Egress-ARP
- L3-Egress-FEC
- L3-ECMP
- L2VPN
- L3VPN
- MPLS
- VPLS
- QOS
- VPWS

IP-Tunnel
Protection (Ingress, FEC, Egress, 1:1, 1+1)
VXLAN
FCOE
EVPN
PON general feature except subnet anti-spoofing
PON MAC limit
TRILL
L2GRE
802.1BR
ROO, ROO VPLS, ROO VXLAN
EVPN
VPLS protection - 2 levels
KBP-FWD
KBP-ACL
MIM/SPB
Hashing (Configured and Flexible)
Egress-FRR additional MPLS label (in both regular MPLS case and EVPN case)
Egress MPLS additional labels (up to 6 MPLS labels same solution as Jericho)
Native-VLAN-Editing: Bridging into VPLS tagged mode tunnel
Support for additional VLAN tag TPIDs
Ingress VLAN considers meter results
MPLS imposition enhancements
3 Ethernet VLAN tags parsing
ROO with native IPv6
Legacy Ethernet OAM (802.1ag or Y.1731)
Legacy Y.1731 over MPLS-TP (GAL), over PWE (ACH)
Legacy BFD over IPv4, over IPv4 over LSP, over PWE (ACH)
Legacy BFD CC over MPLS-TP (GAL)
OAM/BFD statistics
Basic LM/DM
OAM Punt packets
OAM Automatic RDI assertion

Section 4.5.3: Packet Processing New Features Roadmap Plan

Feature/Bug fix	SDK JIRA	Roadmap item	Documentation	Timeframe & Notes
Bridging Into PWE Tagged Mode Tunnel with Service Delimiting	SDK-92490	Native-VLAN-Editing: Bridging into	8847X-AG302-R document, section Deeper	6.5.5 (Done)

Tags		VPLS tagged mode tunnel	Encapsulation Command Stack, example Bridging Into PWE Tagged Mode Tunnel with Service Delimiting Tags	
IP UC Routing Into VXLAN Tunnel	--	ROO VXLAN	8847X-AG302-R document, section Deeper Encapsulation Command Stack, IP UC Routing Into VXLAN Tunnel	6.5.4 (Done)
Segment Routing, IP UC Routing Into Six Deep MPLS Tunnel	--	Egress MPLS additional labels (porting from Jericho/QMX) (see timeframe note)	8847X-AG302-R document, section Deeper Encapsulation Command Stack, Segment Routing, IP UC Routing Into Six Deep MPLS Tunnel (L3VPN Into Deep MPLS Stack)	Not planned, instead, SDK offers the same solution as 88670/88370, for more information see “Egress MPLS Additional Labels” section in document 88670-PG11X-R , porting is planned in 6.5.5 (Done)
Native Ethernet Encapsulation Stage	--	Native-VLAN-Editing: Bridging into VPLS tagged mode tunnel	8847X-AG302-R document, section ETPP Native Ethernet Encapsulation and Native Ethernet Editing Stage	6.5.4 (Done)
Native Ethernet VLAN Editing Stage {VSI, Global-OutLIF}	--	Native-VLAN-Editing: Bridging into VPLS tagged mode tunnel	8847X-AG302-R document, section Native Ethernet VLAN Editing Stage	6.5.4 (Done)
Native Ethernet VLAN	--	--	8847X-AG302-R	Not planned

Editing Stage {VSI, Global-OutLIF.Profile.Domain)			document, section Native Ethernet VLAN Editing Stage	
Native Ethernet VLAN Editing Stage default EEDB entry	--	Native-VLAN-Editing: Bridging into VPLS tagged mode tunnel	8847X-AG302-R document, section Native Ethernet VLAN Editing Stage	6.5.4 (Done). Note: Only 2 fixed options are available in SDK for default EEDB entry.
Three Ethernet VLAN Tags Parsing	SDK-89417	3 Ethernet VLAN tags parsing	8847X-AG302-R document, section Three Ethernet VLAN Tags Parsing	6.5.5 (Done)
Support for Additional VLAN Tag TPIDs	--	Support for additional VLAN tag TPIDs	8847X-AG302-R document, section Support for Additional VLAN Tag TPIDs	6.5.4 (Done)
MPLS Termination After IP-Tunnel Termination	--	--	8847X-AG302-R document, section, MPLS Termination After IP-Tunnel Termination	Not planned
Ingress VLAN Edits PCP-DEI to Include Meter Result Information	SDK-73606	Ingress VLAN considers meter results	8847X-AG302-R document, section, Ingress VLAN Edits PCP-DEI to Include Meter Result Information	6.5.4 (Done)
TPID transparent EVE	SDK-73610	VSWITCH	8847X-PG1XX-R document, section Basic VLAN Translation	6.5.4 (Done)
Imposition of Special Labels	--	MPLS imposition enhancements	8847X-AG302-R document, section MPLS Imposition Enhancements	6.5.4 (Done)
Independent TTL and EXP Inheritance Models	--	MPLS imposition enhancement	8847X-AG302-R document, section MPLS Imposition	6.5.4 (Done)

		s	Enhancements	
ITPP Network Headers Termination (Forwarding Copy)	--	None	8847X-AG302-R document, section, Ingress Transmit Packet Processor Enhancements	6.5.4 (Done)
ITPP Fallback2bridge	--	--	8847X-AG302-R document, section, Ingress Transmit Packet Processor Enhancements	Not planned (see errata issue EID#15 in 8847X-ES101-R)
KAPS Database Hit Bit per Entry	SDK-75711	KAPS DB hit bit per entry	8847X-AG302-R document, section, KAPS Database Hit Bit per Entry	6.5.8
Routing enablers first my-mac	--	L3 (v4, v6)	Will be documented in the next UM version (88470-PG1XX-R)	6.5.4 (Done), note that the feature changes the default behavior of QAX compared to Jericho (see “Backward compatible important notes”)
L2CP functionality per OutLIF	SDK-75719	VSWITCH	Will be documented in the next UM version (88470-PG1XX-R)	6.5.6 (Done)
OAM MEP-DB - Additional data may store Loss Measurement (LM)/Delay Measurement (DM) statistics	--	LM/DM	88470-AG100-R, section Database Enhancements 8867X_8847X-AG10X-R section Adding Delay/Loss Measurement to Accelerated MEP	6.5.4 (Done)
OAM 48B flexible Maintenance	SDK-98807	Fully flexible MAID	88470-AG100-R, section Database	6.5.7 (Beta version in 6.5.6 -

Association Identifier (MAID)			Enhancements	Done)
OAM Unicast continuity check message (CCMs)	--	--	88470-AG100-R, section Database Enhancements	Not planned
BFD jitter	--	--	88470-AG100-R, section New features	Not planned
BFD Authentication	--	--	88470-AG100-R, section New features	Not planned
OAM/BFD Additional TLVs on CCM/BFD packets	--	--	88470-AG100-R, section New features	Not planned
OAM/BFD On-Demand TX Machine	SDK-73623	On demand delay measurement, loss measurement (new), BFD in demand mode (New)	88470-AG100-R, section New features	Not planned
OAM Delay Measurement Statistics Enhancements	--	--	88470-AG100-R, section New features	Not planned, SDK supports only two-way delay.
OAM Configurable Opcodes Have Their Own Maintenance Domain (MD) Levels	--	--	88470-AG100-R, section Bug Fixes and Improvements	Not planned. SDK supports AIS the same as Jericho solution. LCK is not planned.
OAM Punted Packets Include the Source of Failure	--	Punt packets	88470-AG100-R, section Bug Fixes and Improvements	6.5.4 (Done)
OAM RFC 6374	--	--	88470-AG100-R, section Bug Fixes and Improvements	Not planned

OAM Configurable Transmission Rates for Y.1731 messages	--	--	88470-AG100-R, section Bug Fixes and Improvements	Not planned. SDK solution is the same as Jericho (Static values, not configured)
OAM Hierarchical-LM per MD-level (2 levels) – B0 only	SDK-75703	Hierarichical LM: Per MD-Level (2 counters) (New)	TBD	6.5.7 (beta version in 6.5.6)
OAM Hierarchical-LM per LIF (2 levels) – B0 only	SDK-75703	Hierarichical LM: Per LIF (New)	TBD	6.5.7 (beta version in 6.5.6)
OAM RDI automatic assertion for multipoint services	SDK-75726	Automatic RDI asseration	TBD	6.5.5 (Done)
OAM separate report mode for LM and DM	SDK-88515	LM/DM	TBD	6.5.6 (Done)
OAM Downmep injection new mode (Egress PP)	SDK-73593	LM/DM, ETH OAM	Will be documented in the next UM version (88470-PG1XX-R)	6.5.7 (6.5.6 in Beta version)
BFD VCCV Type 3	--	Legacy BFD over IPv4, over IPv4 over LSP, over PWE (ACH)	--	6.5.4 (Done)
BFD your-discr=0 trap	SDK-75728	Legacy BFD over IPv4, over IPv4 over LSP, over PWE (ACH)	--	6.5.6
Validity Checks May Be Disabled per MEP Profile	--	--	88470-AG100-R, section Bug Fixes and Improvements	Not planned, SDK does not expose this functionality explicit, but per feature requirement (for

				example 48B MAID will disable MAID verification)
Increase Punt profiles to 16	--	Punt packets	88470-AG100-R, section Database Enhancements	6.5.4 (Done)
Loss Measurement (LM) and Synthetic Loss Measurement (SLM) Coexist – per LIF decision	SDK-75710	LM or SLM per LIF basis (New)	88470-AG100-R, section Bug Fixes and Improvements	6.5.7
OAMP OAM over PWE injection with 2 OutLifs (including PWE OutLif)	SDK-88994	LM/DM, Legacy Y.1731 over MPLS-TP (GAL), over PWE (ACH)	Will be documented in the next UM version (88470-PG1XX-R)	6.5.7 (6.5.6 in Beta version)

Section 4.5.4: Major Bug fixes

Packet Processing:

See BCM88670-Family major bug-fixes section
When creating OAM/BFD endpoints without explicit MEP-ID, the allocated ID may exceed the allowed 0-8191 range.
OAM - When endpoints are present on both NNI interface - PWE(Down MEP) and UNI interface(UP MEP), UP MEP transmitted packet is trapped DOWN MEP instead of getting forwarded.

Section 4.5.5: Known issues

Network Interface:

Only the interfaces indicated above are validated. If any interface that is currently not supported is blocking initial bring-up, please consult with Broadcom AE.
Auto-training and Auto-negotiation protocols are not validated at this stage
SAT is not functional
Control message of Link Bonding is not supported
Policer/Meter is not functional , depended on CGM block which its work is not completed
Ingress CosQ: when headroom is enabled for a VoQ, it will not be able to use shared memory resources. Therefore it is recommended not to enable headroom until the issue is fixed.

Packet Processing:

OAM delay measurement using 1DM is not functional.
OAM: In response to SLM sent to the MEP, the MEP sends a valid SLR and an additional punt packet. Root-cause: SLM sub-type was changed to 7. There is no HW support for sub-type 7 (SLM) in OAMP. Side effect: SLM/SLR are not supported by OAMP in this release.
ECN-PP is not functional
PON subnet anti-spoofing
KBP-ACL: When using the external TCAM (KBP) for ACLs, the SIP DIP sharing functionality is not available. Thus, the SIP and DIP are invalid qualifiers for KBP ACLs

Section 4.5.6: BCM88470-Family B0 revision support

BCM88470-Family B0 revision introduces bug fixes. For full information see:

8847X-ES10X-R document (BCM88470 Device Errata), summary list

BCM88470-Family B0 revision support is first introduced in 6.5.6. 6.5.6 version will include all B0 registers changes.

BCM88470-Family B0 revision is in the same quality as BCM88470-Family A, same regression tests pass.

Systems containing BCM88470-Family-B0 should remove the A0 suffixes from existing SOC properties which are relevant for those devices. The only exceptions are SOC properties which are specific per revision, In those the revision suffix (A0/B0) should remain.

BCM SDK, on the initialization stage, will enable by default all bug fixes as well as its improvements. This is true as long there are no major behavior expectations. In case there is such a gap, we will emphasize and document such cases.

Following is the list of the expected bug fixes and enhancements that require SDK support and its timeframe.

Feature/Bug fix	SDK JIRA	Documentation	Timeframe	Comments
SPB/MIM ROO MC	SDK-92756	TBD	6.5.6 (Done)	
Native-LL Indication	SDK-105117	TBD	6.5.6 (Done)	
Egress (ERPP) IP/MPLS Protocol Filters are Not Disabled For Control	SDK-105095	TBD	6.5.6 (Done)	

Packets				
NIF – ILKN Link Active Indication Not Correct For Small Ports	SDK-105097	TBD	6.5.6 (Done)	
Wrong forwarding code after MPLS Special Label Termination	SDK-105089	TBD	6.5.7	
MPLS Preprocessing key construction does not take into account consecutive MPLS special labels	SDK-105090	TBD	6.5.7	
MPLS Tunnel-Termination does not consider MTSE0 value in PHB and remarking	SDK-105092	TBD	6.5.7	
IHP – UDP checksum of value 0 is marked as needed to be update	SDK-105091	TBD	6.5.7	
ECMP LB Key0 and ECMP LB Key1 cannot be different when using configured LB logic	SDK-105087	TBD	6.5.7	
PHP doesn’t consider BOS when deciding on next-protocol	SDK-105085	TBD	6.5.7	
Different Location of IP Header Fields in Egress PMF for bridged packet and	SDK-105088	TBD	6.5.7	

IP-MC Fallback-To-Bridge packets				
Hierarchical LM	SDK-75703	TBD	6.5.8	

Section 4.5.7: New Features

See BCM88670-Family new features section, in addition

Basic data path, connectivity and Traffic Management features:

Packet Processing:

Supporting OAM packets with OAM My-CFM MAC identical to L3 My-MAC (also for BCM88680 families)

Y1731oPWE with GAL has been supported with new endpoint Type bcmOAMEndpointTypeBHHPweGAL

Diagnostics:

diag oam counter command can be used to dump LM counters using lif and direction.

Section 4.6: BCM88670 Family GA Release

This release is for the BCM88370-Family and BCM88670-Family product lines.
In the continued SDK support, all features introduced in SDK 6.5.5 are also supported in SDK 6.5.6.
The subsequent sections describe the increment in available features compared to 6.5.5, major bug-fixes and known issues.
It is extremely important to review “Backward compatible important notes” section before starting the integration of the new release.

Section 4.6.1: Backward compatible important notes

SW compatibility guidelines 6.4.X to 6.5.X:

Preserving SW interface of BCM88670 device between SDK versions was a key consideration in 6.5.x device driver design. The SW interface was modified in the following cases: Improved SW design resulting in a more intuitive API, efficiency, significantly modified or extended device functionality.

To assist the migration process from existing application configuring BCM88670 devices over 6.4.x SDK versions (6.4.10, 6.4.11), to an application configuring BCM88670 devices over 6.5.x SDK versions, a dedicated document was created: "BCM88670 BCM SDK Compatibility Guidelines 6.4.x to 6.5.x”. We highly recommend going over the

document. In case you don't find the document in docsafe, please approach your AE engineer to get a copy of it.
Customers that migrate existing applications over 6.4.X releases that are prior to 6.4.10, are required to do two steps: first to go over the release notes of later 6.4.X releases up to 6.4.10 (including) and fit their application configuration accordingly and then start the migration process from 6.4.X to 6.5.X.
ISSU is NOT supported from 6.4.X to 6.5.X.

BCM SDK is aligned to KBP SDK version 1.4.7 (true also for BCM88470 and BCM88680 families)

OAM/BFD with UDH:
UDH (user defined header) is used for Cascading Ingress-Egress lookup in Field-Processor (For more information see PG PP document). This applies only when the soc property field_class_id_size_0/1/2/3 is set.
In order to handle better UDH in OAM/BFD applications, the behavior of UDH included or not in system headers that are trapped to CPU/External OAM host may changed compared to previous releases (6.5.4, 6.4.X):
Down MEP: UDH will always be included except for DMM/DMR packets. These packets may be identified by the existence of an OAM-TS header and by seeing that MEP-Type =1 and OAM-Sub-Type = 2 or 3.
Up MEP: UDH will be included
OAMP Punt packets: the inner set will not include a UDH.
Note that in 6.5.4 the UDH did not appear on trapped Up MEP packets and now in 6.5.5, 6.5.6 it appears.
Note that the change is first introduced in 6.5.5.
The change was done also for BCM88660, BCM88470 and BCM88680 families.

MPLS PORT: SDK-103774 An error has been set, forbidding the setting of mpls_port.egress_label in ingress only mode with the flag BCM_MPLS_PORT2_LEARN_ENCAP. This is because in this case, we learn the encap id, so the VC label should be empty. The change was done also for BCM88660, BCM88470 and BCM88680 families.

IPv6 public RPF: SDK-90144 L3 IPv6UC, public routes(vrf=0) in the KAPS (LPM) are no longer enabled by default. The flag BCM_L3_INGRESS_GLOBAL_ROUTE should be passed to bcm_l3_ingress_create to enable public searches per rif. If the in-rif is associated with vrf=0 and public DB is enabled (public_ip_frwrd_table_size != 0), it must have this flag to enable searches in KAPS. If a single in-rif is associated both with public searches and rpf, the KAPS will perform a total of four searches, private/public forwarding/RPF. Note that this change is aligned to L3 IPv4UC. The change was done also for BCM88470 and BCM88680 families.

Load-balancing: SDK-104145 Added a verification in bcm_switch_control_set with switch_type = bcmSwitchTrunkHashConfig, bcmSwitchECMPHashConfig and bcmSwitchECMPSecondHierHashConfig to prevent same polynomial configuration for more than one hash index (ECMP, LAG or consistent). The correct configuration should have different hash polynomial configuration for each load-balancing type. The change was done also for BCM88660, BCM88470 and BCM88680 families.

EVPN: SDK-105215 In EVPN application, an error has been issued in case an IML label is configured vis bcm_mpls_tunnel_switch_create with flag "BCM_MPLS_SWITCH_EVPN_IML" with no pre configuration of a label range via bcm_mpls_range_action_add(). The change was done also for BCM88680 and BCM88470 families.

MPLS PORT: SDK-104875 When calling bcm_mpls_port_add, in case criteria parameter equals BCM_MPLS_PORT_MATCH_INVALID, match_label cannot be set. This is now enforced. The change was done also for BCM88660, BCM88680 and BCM88470 families.

SAT: SDK-100384 In previous version, high threshold(kbits) and low threshold(kbits) were wrongly set to HW, which caused an issue of starting traffic at wrong threshold in simple burst mode. High threshold and low threshold can be set via bcm_sat_gtf_rate_pattern_set. After the fix, above issue is fixed. In Simple Burst mode, SAT engine collects credits until it reach high threshold and then starts sending traffic, until credit balance is below low threshold. In Continues mode, high threshold and low threshold should be same, and the values for high threshold and low threshold are suggested to be a small value. The change was done also for BCM88680 and BCM88470 families.

IPMC: SDK-104506 when adding bcm_ipmc_add with flag BCM_IPMC_IP6, group field is not updated as before. In previous releases group field was changed to remove the L3 multicast group bit indication from the group field. This change was removed in 6.5.6 and now the group field keeps the L3 multicast group bit indication as is done in IPv4 MC group. The change was done also for BCM88660, BCM88680 and BCM88470 families.

COSQ: SDK-104976 Multicast traffic replicated in the fabric using standard fabric MC (4 Fabric Multicast Queues, FMQs) was enqueued into FMQ-0 regardless traffic TC. This was fixed, to distribute MC traffic to FMQs 0-3 according to TC.

Pay attention that if you used this function and assumed all queues are unicast queues, you might get errors.

Section 4.6.2: New Features

Basic data path, connectivity and Traffic Management features:

ILKN non-consecutive lanes support

LinkScan: The link status was only checked and determined by PHY status, which was not enough. Local/Remote fault checking has been added in Linkscan thread in this release.

PHY MDIX: New API bcm_port_mdix_set is added to support external PHY MDIX configuration.

Packet Processing:

MPLS-TP Section OAM (also for BCM88470, BCM88680 families) Supported endpoint type bcmOAMEndpointTypeBhhSection.

Following items are the supported use cases:

CCM Processing by OAMP.

CCM generation from OAMP.
LMM packet stamping by OAM classifier
LMM generation from OAMP
LMM processing by OAMP and LMR generation.
DMM generation from OAMP
DMM processing by OAMP and DMR generation.
SLM generation by OAMP
Following items are not supported in section-OAM:
LBM generation
LBM processing and LBR generation
SLM Processing and SLR generation by OAMP
Hierarchical LM (currently only in PRE stage) - only for BCM88470-B0

IP-Tunnel enable encapsulation of GRE-key LB-information (also for BCM88470, BCM88680 families)
IP-Tunnel termination - added port_class (PP port mapping) to tunnel termination mode 6 (DIP,SIP,VRF) - (also for BCM88470, BCM88680 families)
VPWS tagged mode - added the support of two lookups for Inner ETH double-tagged (2-tags and 1-tag lookups) - (also for BCM88470, BCM88680 families)
Added the support of Stacking extension without enabling stacking application (also for BCM88470, BCM88680 families)

High Availability:

SER Interrupt: For some interrupts, the SER handling would do hard reset or software reset. In this release, an option is added to allow calling user callback instead of doing reset.
SER Interrupt: In previous releases, if an interrupt occurred but not handled, no information was printed. In this release, interrupt info can be printed out for every interrupt routine.
Exact Match Shadowing: SOC property exact_match_tables_shadow_enable supports a new value: 2 - Disabled for LEM, enabled for other exact match tables. Note that MAC table bulk operations are not supported when LEM shadow is enabled.

Diagnostics shell:

PHY Diag: External phy and system interface type were not supported for command "phy diag <unit> dsc". In this release, new arguments are added to support external phy and interface. Command example, "phy diag <unit> dsc u=1 if=sys".
PHY PRBS: PRBS for external PHY was not supported. In this release, this feature is added and new command is added. PHY unit, interface type and poly can be spcecified. Command example, "phy diag <port> prbs u=1 if=sys p=3".
PHY Diag: Added PCS related commands:

“phy diag <port> pcs”
“phy diag <port> pcs topo”
“phy diag <port> pcs link”
“phy diag <port> pcs speed”
“phy diag <port> pcs aneg”
“phy diag <port> pcs tfc”
“phy diag <port> pcs antimers”
“phy diag <port> pcs state”
“phy diag <port> pcs debug”

Section 4.6.3: Major Bug fixes

The list below refers to major bugfixes, and does not provide a comprehensive coverage of various bugfixes on all levels.
Basic data path, connectivity and Traffic Management features:

Fixes for credit request profiles:

The default application configures SCH to be SLOW (previously was LOW_DELAY).

The default application associates each port with its corresponding SLOW credit request profile (previously all ports associated with 10G_SLOW profile).

Introduce fine-tuned credit request thresholds in case of local switch (single device).
Calling bcm_port_loopback_set(unit,port,BCM_PORT_LOOPBACK_PHY) on a disabled port of Falcon interface brought the port to be enabled. This issue is fixed.
After the fix, ports status keeps no change during loopback setting.
KBP: There was no re-transmission mechanism in KBP XPT layer in previous releases. Once transmission failed in XPT layer, it would return error to application layer. And KBP LUT initialization was not stable because of it. In this release, a re-transmission mechanism has been added into KBP XPT layer.
EXT PHY: If external PHY AN result is 1G, system side was down. This caused traffic was blocked. This issue has been fixed in this release.
EXT PHY: 40G ports could not be probed and loopback did not work. These isses are fixed. External PHY 40G mode works proprerly in this release.
EXT PHY: The link status flipped when the external PHY worked on pcs repeater mode. This issue has been fixed in this release.

High Availability:

Fixed incorrect caching settings for some talbes and registers
Below tables are changed to be uncacheable:
KAPS_TCM
SCH_MEM_01F00000

SCH_MEM_30000000
ILKN_PMH_PORT_0_CPU_ACCESS
ILKN_PMH_PORT_1_CPU_ACCESS
ILKN_PML_PORT_0_CPU_ACCESS
ILKN_PML_PORT_1_CPU_ACCESS

Below tables are changed to be cacheable:
PPDB_A_FEC_ENTRY_FORMAT_A
PPDB_A_FEC_ENTRY_FORMAT_B
PPDB_A_FEC_ENTRY_FORMAT_C
PPDB_A_FEC_ENTRY_FORMAT_NULL
PPDB_A_FEC_ENTRY_GENERAL
PPDB_B_LARGE_EM_FORMAT_1
PPDB_B_LARGE_EM_FORMAT_2
PPDB_B_LARGE_EM_FORMAT_3_TYPE_0
PPDB_B_LARGE_EM_FORMAT_3_TYPE_1
PPDB_B_LARGE_EM_FORMAT_SLB_COUNTER
PPDB_B_LARGE_EM_LEARN_FORMAT
IPS_QPM_1_NO_SYS_RED
bcm_petra_rx_init() might return error during parallel initialization. This issue is fixed.
DRCx_PhyCdrAboveTh interrupt might be triggered during soft reset. This bug has been fixed.
In previous releases, some cachable tables didn't initialize properly, led to SER recovery failed at the first time. In this release, this problem is fixed.

Packet Processing:

EVPN: IVE functionality was not executed for EVPN packets (also for BCM88470, BCM88680 families)
L3/IPMC: when adding an entry to the LPM (KAPS), using the BCM_L3_FLAGS2_RAW_ENTRY or BCM_IPMC_RAW_ENTRY respectively, some payload values were rejected due to incorrect verification (also for BCM88470, BCM88680 families)
OAM: OAM packet could be trapped to the CPU because of an outer LIF MEP. Even though there is no MEP on the actual LIF.
for example, PWE OAM packets might be trapped for PWE o LSP o ETH packet while an endpoint is only set on the LSP (also for BCM88470, BCM88680 families)
OAM: When SLM packet is received in the OAMP, the OAMP respond with SLR packet. The SLR packet should be transmitted on the Core where the MEP is configured, SLR was previously transmitted only on core 0 (also for BCM88680 families)
OAM: standard MEP configuration can affect also default MEP configuration. This means that under certain circumstances traps created by the endpoint will remain (also for BCM88470, BCM88680 families)

IP Tunnels: memory is leaked upon continuous update of the same object via bcm_tunnel_initiator_create() (also for BCM88470, BCM88680 families)

Section 4.6.4: Errata

The list below relates to major open bugs that are not resolved:

Basic data path, connectivity and Traffic Management features:

PM4x10 WB isn't supported.

Packet Processing:

OAM UP-MEP default profiles (multiple egress default LIF-profiles) is not functional. Also true for BCM88470, BCM88680 families.

Tunnel IPv6: bcm_tunnel_initiator_clear does not delete global_to_local LIF mapping in case of IPv6 tunnel. Also true for BCM88470, BCM88680 families.

Ingress-PMF Large direct lookup: there is a known issue in the payload formatting, which results in inconsistency in the PMF actions. Also true for BCM88470, BCM88680 families.

Section 4.7: BCM88680 (Jericho+) Family GA Release

This release is a GA version for the BCM88680-Family product line, following previously released bring-up EA versions.

Section 4.7.1: Backward compatible important notes

See BCM88670-Family "Backward compatible important notes" section.

Section 4.7.2: Bring-up guidelines

Reference Documentation

Multiple documents describing relevant HW and SW aspects are available, including:

SW Documentation

88X7X-PG2xx: Traffic Manager Theory of Operation - This document describes theory of operation and provides driver reference for the BCM88670 device series.

88680-PG1xx: BCM-API Packet Processing: Theory of Operation - This document describes BCM-API for the BCM88680 device packet processing capabilities, and how to configure it for networking applications.

BCM88670/BCM88680 Software Compatibility Guide - This document summarizes the differences in the BCM SDK software interface between

BCM88670 and BCM88680 devices. This document is intended for customers using the BCM SDK to configure BCM88670 devices. The goal of this document is to assist the migration process from existing applications configuring BCM88670 devices to an application configuring BCM88680 devices.

HW Architecture specs:

88670-AG2xx: Traffic Management Architecture - This document describes the BCM88670/6BCM88370 and BCM 88680 traffic management architecture and fabric adapter. It is intended for system architects and anyone else seeking an understanding of the features and capabilities that the BCM88680 traffic management architecture provides.

8868X-AG1xx: BCM8868X Packet Processor Architecture Changes - This document is intended for system architects and anyone else seeking an understanding of the features and capabilities that the BCM8868x Packet processing architecture provides.

Diagnostics Shell

The diagnostics commands below are useful for debugging. All the diagnostics are called from the BCM shell. If you type the diagnostic incorrectly, usage will be printed. The following diag shell commands are validated:

General

diag count g - display a graphical representation of the device counters

show counters: MIB counters

clear counters

NIF/Ports:

diag nif - display link status information.

diag port_db (diag port) - display ports information, mapping to PMs, channels etc.

Packet DMA:

"tx" shell command

PP:

diag pp

kbp kaps_show

diag dbal

Port Mapping

Core association must be defined upon initialization, using the new format of BCM8867x:

ucode_port.port<logical-port-id>.<unit> =

<Interface_name>.<channel_num>:core_<core-id>.<tm- port-id>

DRAM Interface

It is recommended to start the bring-up in "SRAM-only" mode. Before trying to use DRAM. SOC properties: ext_ram_present=0 to work in SRAM-only mode.

When working with DRAM, change ext_ram_present=<0|1|21|22|3> according to the number of DRAM interfaces - seed details in the UM (88x7x-PG2xx).

When working with DRAM, please note that for each device, for the first time you need to run DRAM PHY calibration. Once calibration parameters are determined, they can be saved and restored upon later initializations to significantly reduce initialization time. See Driver Reference— DRAM PHY Tuning section in the UM for details.

Network Lane Swap and Polarity configuration

Network Interface
Lane swap and polarity - the direction of configuration for both RX and TX is from front panel to device. For example, for the swap configuration described below the correct swap-mapping is 0x0321.
Device lane0 -> front panel lane3
Device lane1 -> front panel lane0
Device lane2 -> front panel lane1
Device lane3 -> front panel lane2
Programmable ITMH

ITMH (Ingress Traffic Management Header) in 8867x format works.

Section 4.7.4: New Features

Packet Processing:

See BCM88670-Family, BCM88470-Family New features sections

Section 4.7.5: Major Bug fixes

Packet Processing:

See BCM88670, BCM88470 families Major Bug fixes section

Section 4.7.6: Known issues

Packet Processing:

OAMP server is in beta stage.

OAM packet counting issue: There is a known counting issue at the ingress for the BCM88680-family. Packet will not be counted if it’s inner-most recognized LIF is different than the LIF on which a MEP resides. For example, packet of the following structure PAYLOAD o PWE o LSP o ETH, will not be counted for a configured MEP on the LSP LIF if the PWE LIF is configured.

OAM: In response to SLM sent to the MEP, the MEP sends a valid SLR and an additional punt packet. Root-cause: SLM sub-type was changed to 7. There is no HW support for sub-type 7 (SLM) in OAMP. Side effect: SLM/SLR are not supported by OAMP in this release.

Ingress-PMF Large Direct Lookup is not functional due to inconsistency in the payload formatting.

Section 4.8: BCM88770 (FE3600) Release

The Broadcom BCM88770 (formerly named BCM88950) is the fourth generation in the Dune product line of Fabric Element (FE) devices.

This is a sustaining release for BCM88950 driver, with all major features supported.

4.8.1: Important Notes

- The default DCS thresholds were optimized: RCI, GCI, LLFC, Drop.

4.8.2: Major Bugfixes

- None

Section 4.9: BCM88270 (Qumran-UX) Family Beta Release

This release is a beta version for the BCM88270-Family product line, following previously released 6.5.5 Preview version.
The subsequent sections describe the features validated for this release, known issues and bring-up guidelines.
It is extremely important to review “Backward compatible important notes” section before starting the integration of the new release.
Note: IPsec SW package is available as a separate add-on upon request.

Section 4.9.1: Backward compatible important notes

See BCM88670-Family “Backward compatible important notes” section.

Section 4.9.2: Validated Features

Basic data path, connectivity and Traffic Management features:

Register & Memory access including DMA
Supported SKUs:
Qumran-UX: 88270
Device core mode: Single-core

Interfaces:

- SRAM, DDR4@1.6GHz
- CPU RX/TX (packet DMA, RCPU)
- NIF 156.25MHz and 125MHz
- NIF:
  - 1GE: SGMII, GMII, GE
  - 2.5GE: GMII
  - 10GE: KR/XFI, RXAUI, XAUI (Eagle port only)
  - 40GE: KR4, XLAUI

Forwarding:

Unicast
Multicast
mirroring

CosQ:

End-to-end scheduling
VOQ ingress queue creation. Mapping by-destination and by-flow-id
Tail Drop
Egress queuing: number of priorities per port: 1/2/8

Packet forwarding - ITMH, Force-FWD

Packet Processing:

Diagnostics
Port
STG
L2-Forwarding
L2-Learning
VLAN-Translation
VLAN
VSWITCH
L3-IPv4
L3-IPv6
L3-Interface-RIF
L3-Egress-ARP
L3-Egress-FEC
L3-ECMP
L2VPN
L3VPN
MPLS
VPLS
QOS
VPWS
IP-Tunnel
Protection (Ingress, FEC, Egress, 1:1, 1+1)

VXLAN
PON general features except subnet anti-spoofing
TRILL

Section 4.10: BCM88660 (ARAD+), BCM88650 (ARAD) Release

This is a sustain release of BCM88660, BCM88650 driver, with all major features supported.

Section 4.10.1: Important notes

- See BCM88670-Family GA release section, important: SW compatibility guidelines 6.4.X to 6.5.X.

Section 4.10.2: Major Bugfixes

- COSQ: SDK-104976 Multicast traffic replicated in the fabric using standard fabric MC (4 Fabric Multicast Queues, FMQs) was enqueued into FMQ-0 regardless traffic TC. This was fixed, to distribute MC traffic to FMQs 0-3 according to TC.

Section 4.10.3: Errata

None

Section 4.11 New External PHY Support

Section 4.11.1 BCM82332 (Dino) added support

The following are updates from the last release for this device. Note that features have been tested with the BCM56860 switch device:

Speeds/Modes

1G KX,CX, SGMII

HIGIG IEEE Modes

10G: KR, CR, XFI
40G: KR4, CR4, XLAUI

Current device limitations in this release

Warm boot is not fully supported.
When Flexing from 100G to 10G, port counters cannot be cleared
100G GB mode is not functional when system side lanes are selected between 1 to 10 or 2 to 11.
100G PT per lane control on line side is not supported

Digital loopback on line side is failing with 1G.
BER on Merlin lanes is not supported
AN for 100G CR10 PT and 100G HiGIG CR4 GB not supported in this release.
Port detach using API is failing
Digital loopback is failing in HiGIG 42G
Inconsistent TX Lane squelch behavior between Falcon and Merlin lane cores

Section 4.11.2 BCM54190 support

Preview support has been added for the BCM51490 external PHY driver in this release. Please refer to the API reference guide 56XX-PG656-R for more details about changes in this release.

Section 4.11.3: Interoperability testing for new PHY/switch combinations

The following switch and PHY combinations have been interoperability tested in the SDK 6.5.6 release. Below lists only the features that have been tested and supported in this release.

Section 4.11.3.1: BCM56565 and PHY BCM54616S

Features supported:

- 10M/100M/1G Copper
- 100M Fiber
- 1G is supported in Auto-Negotiation mode only
- Diagnostic
  - PRBS, DSC dump, Eyescan and TX Squelch are not supported
- Limitations
  - PCS Loopback is not supported in this release

Section 4.11.3.2: BCM56565 and PHY BCM84756

Features supported:

- 10G Forced Speed mode only
- Diagnostics
  - Line side PRBS, DSC dump, PCS/PMD loopback and Eyescan are not supported

Section 4.11.3.3: BCM56760 and PHY BCM54180/BCM50180

Features supported:

- 10M/100M/1G Copper
- 1G is supported in Auto-Negotiation mode only
- PCS Loopback
- Diagnostic
  - PRBS, DSC dump, Eyescan and TX Squelch are not supported

Software support for EEE (Energy Efficient Ethernet) with BCM56760 switch is not supported in this release.
Support for 1588 is not present in this release.

Section 4.11.3.4: BCM56760 and PHY BCM54380

Features supported:

10M/100M/1G Copper

1G is supported in Auto-Negotiation mode only

PCS Loopback

Diagnostics - PRBS, DSC dump, Eyescan, TX Squelch not supported

Software support for EEE (Energy Efficient Ethernet) with BCM56760 switch is not supported in this release.

Section 5: Things to note

This section lists items that require special attention that are new to this release. Please see prior 6.5.x release notes for a list of older items that should also be noted.

Section 5.1: External PHY Driver SDK Support

Starting in CY3Q16, Broadcom will be providing only discrete drivers for new external PHY devices. Drivers for new external PHYs will not be integrated in the SDK. Broadcom will support standalone drivers for External PHYs which customers can integrate into their software stack. Legacy PHY driver code that has been tested against legacy switch devices will continue to be supported.

Section 5.2: Policy on API defaults

Broadcom does not guarantee default values set within the SDK and changes to default values may be made between releases. If a default value is required for application software to work properly, it must be explicitly set.

Section 5.3: Default config.bcm change from lossless to lossy

In this release, the default MMU setting has been changed from the standard lossless default to lossy in config.bcm for BCM56760 family devices. In SDK 6.5.5, the default was lossy for BCM56270, and BCM56565 device support. Devices with GA support in prior SDK releases remain with the lossless default. This will be the direction going forward for all new devices, while legacy devices default will remain as is.

Section 5.4: ALPM enhancements for BCM56960, BCM56860, and BCM56850

In this release, enhancements have been made to increase the ALPM utilization to reach a higher table capacity.

For XGS devices using the ALPM feature such as BCM56850, BCM56860, and BCM56960, the ALPM utilization has been improved through introducing bucket merge and bucket repartition. As a result, the guaranteed route capacity for IPv4 and IPv6 in ALPM mode is now increased.

Section 5.5: BCM56960 - Mix of Autoneg and Forced Speed Ports within a single Falcon Core is not supported

Due to a single PLL constraint inside the Falcon, a mix of both autoneg and forced speed on a single core is currently not supported. All the ports on a single core should be all in autoneg or forced speed mode. The first port of a core doing autoneg will disable the rest of the forced speed ports on the same core. Once the first port on a core completes autoneg and the VCO is locked, the rest of ports on the same core can only do autoneg with the same VCO.

Section 5.6: Change in behavior of bcm_port_autoneg_set API for BCM56565, BCM56760, BCM56860 device family

On BCM56565, BCM56760, and BCM56860 device families, advertising a speed, on a port, that is not supported by the port block (because of the PLL parameters) may result in a failure during configuring the port to the resolved speed even if auto-negotiation succeeds. There are also other restrictions in the chip for these class of chips (e.g., different ports within a port block cannot have different speeds when the port block is configured in oversub calendar) which can cause reconfiguring the port to fail.

In such a case a switch event BCM_SWITCH_EVENT_AUTONEG_SPEED_ERROR will be generated to notify the user of the failure. It is expected that application software either disables autoneg or remove the failing speed from the advertisements and restart autoneg.

Section 5.7: Change in behavior of bcm_port_speed_set API for BCM56565, BCM56760, BCM56860 device family

On 56565, 56760 and 56860 device family, setting a speed, on a port, that is not supported by the port block (because of the PLL parameters) may result in a failure during configuring. There are also other restrictions in the chip for these family of devices (e.g., different ports within a port

block cannot have different speeds when the port blockis configured in oversub calendar) which can cause reconfiguring the port to fail. In such a case it is recommended to use the bcm_port_resource_multi_set API to update all members of the port block simultaneously.

Section 5.8: Recent PHY interface library changes

The 100G interface for BCM56860, BCM86375, and BCM88560 devices required extensive changes to the interface library called "Portmod". Between SDK 6.5.2 and 6.5.3, BCM5686x operations for "Portmod" with BCM8279x family was verified and BCM8279x has achieved GA quality. For legacy 10G/40G devices, the "Portmod" changes required the shim layer to be created to enable the drivers to connect to the Falcon and Eagle cores. This work was completed for SDK 6.5.3, thus making these legacy PHYs operational with the Portmod capable switches noted above.

Section 5.9: Config variable changes for portmod devices in 6.5.6

In this release, several changes were made for portmod SOC properties as outlined below.

Section 5.9.1: Polarity flip

New configuration variables (SOC properties) were added to flip the polarity for rx and tx lanes.

phy_chain_rx_polarity_flip_physical{<phys_port>.<phy_num>} = value (can be 0 or 1)
phy_chain_tx_polarity_flip_physical{<phys_port>.<phy_num>} = value (can be 0 or 1)

The above variables can only accept Physical port number - corresponding to a physical lane within serdes core or an external Phy.

Section 5.9.2: Lane mapping

New configuration variables (aka SOC properties) were added to specify the lane map on rx and tx lanes.

phy_chain_rx_lane_map_physical{<phys_port>.<phy_num>} = value
phy_chain_tx_lane_map_physical{<phys_port>.<phy_num>} = value

where value is a 16 bit number where each nibble represents the lane mapping for the lane:

bit3-0: lane 0 of physical port <phys_port>
bit7-4: lane 1 of physical port <phys_port>
bit11-8: lane 2 of physical port <phys_port>
bit15-12: lane 3 of physical port <phys_port>

Section 5.9.3: Changes to existing Polarity flip SOC properties

The following SOC properties will not support { } physical port notation starting from this release.

phy_rx_polarity_flip

phy_tx_polarity_flip

Section 5.10: Engineering support statement for older XGS SDK releases

The following releases are now out of XGS engineering support since they are older than 12 months since release:

SDK 6.4.x releases: 6.4.7, 6.4.6, 6.4.5, 6.4.4, 6.4.3, 6.4.2, 6.4.1, 6.4.0
All SDK 6.3.x and older releases

Service impacting defects will be reviewed and potentially addressed on these older releases. Support for non-service impacting defects will only be provided to customers running a SDK release that is less than 12 months old.

The following table shows the number of issues and improvements that have been added to our supported SDK releases by device over the past 12 months. While the table shows individual devices, many issues and improvements will apply to multiple products, e.g. BCM56850 and BCM56860, or all XGS products.

Table 4: Resolved issues and improvements per older XGS device family

Device	***Bugs resolved in the last 12 months***	***Improvements added in the past 12 months***
BCM56960 family	250	58
BCM56850 family	221	41
BCM56840_PLUS	31	3
BCM56340 family	44	12
BCM56640 family	79	17
BCM56450 family	127	30
BCM56150 family	14	0
BCM56440 family	45	16
BCM53440 family	35	4

Section 5.11: Warmboot Notes and Considerations

Section 5.11.1: Validated Warmboot upgrades

Following warmboot upgrades in the table below have been validated in this release.

Software Upgrade	*Supported*
6.4.11 to 6.5.6	Yes
6.5.5 to 6.5.6	Yes

Warmboot testing and issue resolution has focused on the following families of devices:

BCM53400/BCM56060
BCM56150
BCM56220
BCM56340
BCM56440
BCM56450
BCM56640
BCM56840
BCM56850
BCM56960
BCM56260
BCM56460
BCM56965
BCM56160
BCM56760
BCM56565
BCM56270

Section 5.11.2: Validated Warmboot downgrades

Broadcom has performed limited release downgrade testing from 6.5.6 to 6.5.5 and 6.5.6 to 6.4.11 for BCM56960 and BCM56850 family devices.

Portmod does not support warmboot downgrade on BCM88670, BCM56860, BCM56565, and BCM56760 device families.

Section 5.11.3: Determining warmboot scache requirements

The warmboot scache size requirements for a device for a particular release can be found by running the command "warmboot storage" at BCM prompt.

Specifically, to avoid warmboot failure due to out of memory the BCM56565 and BCM56760 family of devices require 6MB size of scache memory.

Section 5.11.4: Warmboot upgrade considerations

For BCM56860 and BCM56960 device support, the map between multicast nexthops maintained implicitly in SDK and L3 interface was not restored in the previous release, this info might be incorrect and cause issues after warm boot. Patches from SDK-94122 and SDK-100058 should be applied to current version of SDK before upgrading from 6.4.11 to avoid this issue.

For BCM56450 devices, warmboot upgrade from 6.4.11 to 6.5.6 or 6.5.3 to 6.5.6 will require a patch from SDK-96015 to be applied to upgraded from version before starting upgrade.

Scache files for BCM56565 family devices generated by SDK 6.5.5 and SDK 6.5.6 will not update the warmboot version number properly. If planning to upgrade from 6.5.5 to 6.5.6, please patch the fix in SDK-106135 to 6.5.5 before upgrading to 6.5.6.

Section 5.11.5: Warmboot support for Embedded Applications

This section describes warmboot support and procedures for associated firmware supported in SDK 6.5.x. Below are the various embedded applications (FW) available and supported with SDK 6.5.x:

- BFD
- BHH LM/DM
- MPLS LM/DM
- ETH LM/DM
- Broadsync
- 1588/PTP

All applications support warmboot with the exception of 1588/PTP. Warmboot for the above apps follows the same process like any other module on SDK side. Eapps warmboot support is through Level-2 warmboot i.e. requires information to be stored in sCache.

SDK warmboot doesn't affect execution of embedded applications (except 1588/PTP firmware) on internal ARM cores and they keep running as is and maintain their state.

Warmboot procedures

- Users are expected to invoke "sc ControlSync=1" command in the BCM diag-shell to store the information in sCache
- The corresponding API is bcm_switch_control_set(int unit, bcm_switch_control_t type, int arg)
    - Note: Typically, storing of information in sCache is required when there is a module or session or context creation on SDK side. Calling of the bcm_switch_control_set() API is left to the discretion of user/customer depending on their system requirements for example, after each session creation or module init or before the warmboot procedure initiation
- Free up the resources attached to the unit by calling below API equivalent of "exit clean" BCM shell command. i.e, "soc_shutdown (int unit)"
    - Note: Broadsync warmboot procedure mandates usage of the above API. As part of API, SDK informs the embedded ARM core about the warmboot process initiation so that BroadSync FW waits for the SDK warmboot procedure to complete for further communication

Steps to follow during the SDK warmboot

Below is an example sequence of diag-shell commands executed (through reload.soc) as part of the SDK warmboot sequence for initializing SDK HOST and firmware communication:

MCSMsg INIT;
MCSMsg 0;
MCSMsg 1;
"MCSMsg INIT" and "MCSMsg x" commands invokes the below APIs respectively:
soc_cmic_uc_msg_start(int unit) /* Allocates Messaging resources/Mutexes */
soc_cmic_uc_msg_uc_start(int unit, int uc) /* Starts Messaging with specific core.*/

Customers will need to include the above calls in their software.

Section 5.12: Release notes for NIF in BCM88675, BCM88375, BCM88680 and BCM88470 families.

The below list refers to release notes for the NIF side of the DNX 88675, 88375, 88680 and 88470 products. Unless otherwise specified – each bullet is relevant for all devices included in these families.

1. **Usage of interface_type with ILKN ports**: When defining an ILKN port, interface_type should not be defined, neither statically (with serdes_if_type SOC property) nor dynamically (using bcm_port_interface_set API). 6.5.6 version will return an error for any such attempt. Configuring the relevant phy parameters (such as DFE, LPDFE and media_type) should be done by using their specific APIs.

2. **Default interface_type for XE port in 25G:** Starting on 6.5.6 version, default interface_type of XE in 25G was modified to XFI. Note that FW configuration (DFE, LPDFE, and media_type) of XFI is identical with FW configuration of previous default interface_type (KR).

3. **Configuring SerDes for data speed 10600:** In order to be compatible with XGS convention, the logical name of data rate 10600 was changed to 11000. The data rate of 11000 was already supported in previous versions and was configuring the same SerDes clock as 10600 (SerDes clock of 10.937G) so using data rate of 11000 instead of 10600 will have no actual effect on NIF behavior. However, starting from 6.5.6 version – trying to set data rate of 10600 will return an error for either PM10 or PM25 ports.

4. **Using SW Macro to set NIF Priority Drop MAP key:** When setting NIF Priority Drop, user must use the following SW macros BCM_COSQ_INGRESS_PORT_DROP_TM_MAP_KEY, BCM_COSQ_INGRESS_PORT_DROP_ETH_MAP_KEY, BCM_COSQ_INGRESS_PORT_DROP_MPLS_MAP_KEY (88680 only) and BCM_COSQ_INGRESS_PORT_DROP_IP_MAP_KEY (88680 only) in order to set TM, ETH, MPLS and IP MAP key. Using these macros was always the recommended method for setting the MAP key but starting from 6.5.6 – setting these keys directly by SW code without using the above mentioned macros will cause error when using the MAP key in PRD BCM APIs later on.

5. **Potential traffic loss during Dynamic Port Provisioning:** Temporary traffic loss might happen during the execution of Dynamic Port Provisioning on another port that is sharing the same quad. The root cause is that when dynamically adding a port, 6.5.6 release runs code that temporary cleans the port priority of the entire quad before retrieving the original priorities. This temporary cleaning of the NIF priority leads to temporary traffic loss.

   6.5.7 code has a fix for this issue.

6. **NIF Priority for ILKN ports on QAX:** QAX doesn't support NIF Priority for ILKN ports. NIF priority for ETH ports is supported.

   The above bullet is relevant for 88470 family only.

7. **Setting NIF PLL when using ILKN on QAX:** In QAX, in order to allow loading ILKN port on more than one NBI block, when ILKN is used, we tie the NIF PLLs of PML0/PML1 to take their input from the output of PMH NIF PLL, so all NBIs will use the same PLL. Therefore, when ILKN is used, the SOC property configuration should be: serdes_nif_clk_freq_**out_2** = serdes_nif_clk_freq_**in_0** = serdes_nif_clk_freq_**in_1**.

   The above bullet is relevant for 88470 family only.

8. **bcm_port_get 'interface_info.phy_pbmp' returned value for ILKN ports:** In previous versions (6.5.5 and earlier) the API bcm_port_get was returning interface_info.phy_pbmp in range of 1-72, which didn't match the bcm_port_add API since in the add API, flag BCM_PORT_ADD_USE_PHY_PBMP can be used only for ILKN ports and interface_info.phy_pbmp should be used in the range of 0-23 (similar to soc property ilkn_lanes).

   In 6.5.6 and beyond, interface_info.phy_pbmp will be:

   · For ILKN port: 0-23 range. The flag BCM_PORT_ADD_USE_PHY_PBMP will be set.

   · For non-ILKN ports: 1-72 range.

   See UM for details.

9. **Soc property Ilkn_lane_map is set per logical port:** ILKN specific soc properties are usually per interface id and not per logical port. This is since there cannot be different values per different channels of the same interface. However, ilkn_lane_map soc property is set per logical port. The practical meaning is that it is enough to set this soc property for a specific single channel from the list of channels that are sharing the same physical interface.

Section 6: Summary of BCM API changes and enhancements

This section summarizes BCM API changes in this release. Complete documentation will be available in the Network Switching Software Programmer's Guide number 56XX-PG656-R.

For the full list of API support by Broadcom device, please reference the file SDK-6.5.6-Support-Matrix.xls in the sdk-all-6.5.6/RELDOCS directory in the release package.

Section 6.1: Class of Service Queue Configuration

Added to Flow Control Endpoint flags

Table: Flow Control Endpoint flags

BCM_COSQ_FC_HEADROOM	FC triggered by Sram Headroom Resouce	New

Added to TCP Profile Create Flags

Table: TCB Profile Create Flags

Flag	Description	
BCM_COSQ_TCB_THRESHOLD_PROFILE_WITH_ID	Create a threshold profile with a specified id.	New

Added to Threshold flags

Table: Threshold flags

BCM_COSQ_THRESHOLD_QSIZE_RECOVERY	Queue size recovery threshold settings	New
BCM_COSQ_THRESHOLD_HEADROOM	Set threshold value for sram headroom resource	New

Added to BCM CoSQ gport flags

Table: BCM CoSQ gport flags

BCM_COSQ_GPORT_SCHEDULER_PFC_ALIGN	New

Changes to size flags

Table: size flags

COSQ_GPORT_SIZE_GLOBAL	Configure global values.	Deprecated
COSQ_GPORT_SIZE_DRAM_BOUND	Configure DRAM bound values.	Deprecated
COSQ_GPORT_SIZE_RECOVERY_FAILURE	Configure recovery failure values.	Deprecated
COSQ_GPORT_SIZE_RECOVERY	Configure recovery values.	Deprecated
COSQ_GPORT_SIZE_PACKET_DESC	Configure packet description.	Modified
COSQ_GPORT_SIZE_SRC_VSQ_POOL1	Configure the pool-1 source VSQ, else configure the pool-0 source VSQ.	Modified

Added to bcm_cosq_stat_t

Table: bcm_cosq_stat_t

Name	Description	
bcmCosqStatOBMFlowControlEvents	OBM generated flow control events per port.	New
bcmCosqStatOBMHighWatermark	OBM max usage per port.	New
bcmCosqStatOBMBufferBytes	OBM buffer current usage per port.	New

New Type of resources

Table: Type of a resource.

bcmReservationResourceBufferDescriptors	Resource reservation of buffer descriptors.
bcmReservationResourceOcbBuffers	OCB resource reservation in buffers.
bcmReservationResourceBufferDescriptorBuffers	Resource reservation of buffer descriptor buffers.
bcmResourceBytes	Resource in bytes.
bcmReservationResourceBytes	Resource reservation in bytes.
bcmResourceOcbBytes	Resource in OCB bytes.
bcmResourceOcbPacketDescriptors	Resource in OCB packet descriptors.

New TCB monitor scope type

Table: TCB monitor scope type

Name	Purpose
bcmCosqTcbScopeUcastQueue	Set scope to a unicast queue
bcmCosqTcbScopeIngressPort	Set scope to ingress port
bcmCosqTcbScopeEgressPort	Set scope to egress port

New TCB Control Type

Table: TCB Control Type

Type	Description
bcmCosqTcbControlEnable	Enable or disable TCB.
bcmCosqTcbControlFreeze	Freeze or restart TCB.

New Port direction type

Table: Port direction type.

bcmCosqIngress	Ingress port.
bcmCosqEgress	Egress port.

New Type of FADT threshold

Table: Type of FADT threshold.

bcmCosqFadtDramBound	DRAM bound settings.
bcmCosqFadtDramBoundRecoveryFailure	DRAM recovery failure settings.

Added to CoSQ Control Type Values

Table: CoSQ Control Type Values

Value			
bcmCosqControlEgressUCQueueGroupMinLimitBytes	Minimum gurantee to be allocated for the port's QueueGroup in bytes value.	number of bytes	New
bcmCosqControlEgressUCQueueGroupSharedLimitBytes	Maximum Threshold for the port's Queue Group	number of bytes	New
bcmCosqControlEgressUCQueueGroupSharedLimitEnable	Enable thresholding using port's QueueGroup Shared Limit for this Queue.	value:1 Enable, 0: Disable.	New
bcmCosqControlEgressUCQueueGroupMinEnable	Enable use of port's QueueGroup Guarantee instead of Queue Gurantee for Unicast Queues.	value: 1 Enable, 0: Disable. Note:Cannot be called while there is traffic on the port	New

bcmCosqControlEgressUCQueueGroupSharedDynamicEnable	Enable Dynamic limit for Queue Group threhsold.	value:1 Enable, 0: Disable	New

Added to Resource

Table: Resource amounts used for allocation.

Parameter	Description	
nominal_headroom	nominal amount of the resource in the lossless headroom.	New
shared_pool_fadt	FADT threshold on the amount of the resource in the shared pool.	New

New structure bcm_cosq_fadt_threshold_t

```
typedef struct bcm_cosq_fadt_threshold_s {
    uint32 thresh_min; /* minimum value of the FADT threshold */
    uint32 thresh_max; /* maximum value of the FADT threshold */
    int alpha; /* alpha parameter of the FADT threshold */
    uint32 resource_range_min; /* minimum limit of resource range */
    uint32 resource_range_max; /* maximum limit of resource range */
} bcm_cosq_fadt_threshold_t;
```

New structure bcm_cosq_tcb_threshold_profile_t

```
typedef struct bcm_cosq_tcb_threshold_profile_s {
    uint32 start_threshold_bytes;   /* The start threshold above which
                                       queue start the capture. */
    uint32 stop_threshold_bytes;    /* The stop threshold below which
                                       queue stop the capture. */
} bcm_cosq_tcb_threshold_profile_t;
```

New structure bcm_cosq_tcb_event_t

```
typedef struct bcm_cosq_tcb_event_s {
    uint64 timestamp;           /* timestamp */
    bcm_gport_t ingress_gport;  /* ingress gport. */
    bcm_gport_t egress_gport;   /* egress gport. */
    bcm_gport_t queue;          /* Egress queue gport */
    uint32 queue_size;          /* Queue depth */
    uint32 service_pool_size;   /* Service pool depth */
    uint32 drop_reason;         /* Drop reason vector */
    bcm_color_t color;          /* Packet color */
} bcm_cosq_tcb_event_t;
```

New structure bcm_cosq_tcb_buffer_t

```
/* TCB Pkt Structure. */
typedef struct bcm_cosq_tcb_buffer_s {
    uint64 timestamp;       /* timestamp */
    bcm_gport_t ingress_gport; /* ingress port gport. */
    bcm_gport_t egress_gport;  /* egress egress gport. */
    bcm_gport_t queue;        /* Egress UC queue gport */
    uint32 queue_size;        /* Queue size bytes */
    uint32 service_pool_size; /* Service pool size bytes */
    uint8 buf_data[80];      /* The first 80 bytes of captured packet */
} bcm_cosq_tcb_buffer_t;
```

New structure bcm_cosq_tcb_config_t

```
typedef struct bcm_cosq_tcb_config_s {
    bcm_cosq_tcb_scope_t scope_type;   /* TCB monitor scope type */
    bcm_gport_t gport;                 /* Logical port or GPORT ID */
    bcm_cos_queue_t cosq;              /* Cos queue, reserved field */

    uint32 trigger_reason;             /* Drop event trigger reason */
    uint32 pre_freeze_capture_num;     /* Number of capture
                                          transactions before freeze */
```

```
    uint32 pre_freeze_capture_time;   /* Capture time before freeze,
                                         usec unit */
    uint32 post_sample_probability;   /* sample probability at post
                                         stage, step is 1/16, valid
                                         configure range is 1 C 16, 1
                                         means 1/16 sample, 16 means
                                         sample all */
    uint32 pre_sample_probability;    /* sample probability at pre
                                         stage, step is 1/16, valid
                                         configure range is 1 C 16, 1
                                         means 1/16 sample, 16 means
                                         sample all */
} bcm_cosq_tcb_config_t;
```

New structure bcm_cosq_fadt_info_t

```
/* complementary information for FADT threshold */
typedef struct bcm_cosq_fadt_info_s{
    bcm_gport_t gport; /* target gport */
    bcm_cos_queue_t cosq; /* target cosq */
    bcm_cosq_gport_fadt_threshold_type_e thresh_type; /* threshold type
*/
    bcm_cosq_resource_t resource_type; /* resource type */
} bcm_cosq_fadt_info_t;
```

New APIs: Class of Service Queue Configuration

bcm_cosq_buffer_id_multi_get

Get buffer id associated with a port.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_buffer_id_multi_get(int unit, bcm_gport_t gport,
                                 bcm_cos_queue_t cosq,
                                 bcm_cosq_dir_t direction,
                                 int array_max,
                                 bcm_cosq_buffer_id_t *buf_id_array,
                                 int *array_count);
```

PARAMETERS

unit	(IN) BCM device number
gport	(IN) Logical port or GPORT ID.
direction	(IN) Types for port direction.
array_max	(IN) Number of elements allocated in buf_id_array.
buf_id_array	(IN/OUT) Array of buffer ids.
array_count	(IN/OUT) Actual Number of elements returned in buf_id_array.

DESCRIPTION

Get the associated buffer_id to a specified port or UCQ. There may have multiple buffers associated to one port or UCQ. If the array length exceeds the actual buffer id number, invalid buffer id is filled in exceeding part.

Table: Port direction type.

bcmCosqIngress	Ingress port.
bcmCosqEgress	Egress port.

RETURNS

BCM_E_NONE
BCM_E_XXX

bcm_cosq_fadt_info_t_init

Init function to the structure "bcm_cosq_fadt_info_t".

SYNOPSIS

```
#include <bcm/cosq.h>
void
bcm_cosq_fadt_info_t_init(
int unit,
bcm_cosq_fadt_info_t* info);
```

PARAMETERS

unit	(IN) Unit number
info	(INOUT)

DESCRIPTION

Init function to the structure "bcm_cosq_fadt_info_t".

RETURNS

none

bcm_cosq_gport_fadt_threshold_set

bcm_cosq_gport_fadt_threshold_get

set/get FADT threshold parameters.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_gport_fadt_threshold_set(
    int unit,
    uint32 flags,
    bcm_cosq_fadt_info_t* fadt_info,
    bcm_cosq_fadt_threshold_t* fadt_info)
int bcm_cosq_gport_fadt_threshold_get(
    int unit,
    uint32 flags,
    bcm_cosq_fadt_info_t* fadt_info,
    bcm_cosq_fadt_threshold_t* threshold)
```

PARAMETERS

unit	BCM device number
flags	(IN) flags
fadt_info	(IN) fadt_info (for "_set")
fadt_info	(OUT) fadt_info (for "_get")
fadt_info	(IN) (for "_set")
fadt_info	(OUT)(for "_get")

DESCRIPTION

set/get FADT threshold parameters.

bcm_cosq_tcb_buffer_multi_get

Get packet records in TCB buffer.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tcb_buffer_multi_get(int unit,
                                  bcm_cosq_buffer_id_t buffer_id,
                                  int array_max,
                                  bcm_cosq_tcb_buffer_t *buffer_array,
                                  int *array_count);
```

PARAMETERS

unit	(IN) BCM device number.
buffer_id	(IN) TCB buffer id.
array_max	(IN) Number of elements allocated in buffer_array.
buffer_array	(IN) Array of packet records.
array_count	(IN) Actual number of packet records returned in buffer_array.

DESCRIPTION

Get packet records sampled before and after a packet drop event. This could only be used when TCB is in FREEZE state. When array_max is 0, buffer_array is null, array_count will return the actual record number in TCB buffer.

Packet record format is as below:

```
/* TCB Pkt Structure. */
typedef struct bcm_cosq_tcb_buffer_s {
    uint64 timestamp;       /* timestamp */
    bcm_gport_t ingress_gport; /* ingress port gport. */
    bcm_gport_t egress_gport;  /* egress egress gport. */
    bcm_gport_t queue;        /* Egress UC queue gport */
    uint32 queue_size;        /* Queue size bytes */
    uint32 service_pool_size; /* Service pool size bytes */
    uint8 buf_data[80];      /* The first 80 bytes of captured packet */
} bcm_cosq_tcb_buffer_t;
```

RETURNS

BCM_E_NONE
BCM_E_XXX

bcm_cosq_tcb_cb_register

bcm_cosq_tcb_cb_unregister

Register/unregister a callback function to be called when TCB buffer enter freeze status.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tcb_cb_register(int unit,
                             bcm_cosq_tcb_callback_t fn,
                             void *user_data);
int bcm_cosq_tcb_cb_unregister(int unit,
                       I       bcm_cosq_tcb_callback_t fn);
```

PARAMETERS

unit	(IN) BCM device number.
fn	(IN) Callback function to register/unregister.
user_data	(IN) Arbitrary application value passed to callback function.

DESCRIPTION

Register/unregister a callback function to be called when TCB buffer enter freeze status.

```
typedef void (*bcm_cosq_tcb_callback_t)(int unit,
                                        bcm_cosq_buffer_id_t buffer_id,
                                        void *user_data);
```

RETURNS

BCM_E_NONE
BCM_E_XXX

bcm_cosq_tcb_config_set

bcm_cosq_tcb_config_get

Set/Get the tcb configuration.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tcb_config_set(int unit,
                            bcm_cosq_buffer_id_t buffer_id,
                            bcm_cosq_tcb_config_t *config);
int bcm_cosq_tcb_config_get(int unit,
                            bcm_cosq_buffer_id_t buffer_id,
                            bcm_cosq_tcb_config_t *config);
```

PARAMETERS

unit	(IN) BCM device number
buffer_id	(IN) TCB buffer id
config	(IN/OUT) TCB configuration

DESCRIPTION

Configure or retrieve the TCB (Transient capture buffer) configuration for the specified MMU resource. MMU resource is identified by the buffer_id.

RETURNS

BCM_E_NONE
BCM_E_XXX

bcm_cosq_tcb_config_t_init

Initialize a tcb config structure.

SYNOPSIS

```
#include <bcm/cosq.h>
void bcm_cosq_tcb_config_t_init(bcm_cosq_tcb_config_t *config);
```

PARAMETERS

config	Pointer to TCB configuration structure to initialize.

DESCRIPTION

Initialize a tcb config structure.

```
typedef struct bcm_cosq_tcb_config_s {
    bcm_cosq_tcb_scope_t scope_type;   /* TCB monitor scope type */
    bcm_gport_t gport;                 /* Logical port or GPORT ID */
    bcm_cos_queue_t cosq;              /* Cos queue, reserved field */

    uint32 trigger_reason;             /* Drop event trigger reason */
    uint32 pre_freeze_capture_num;     /* Number of capture
                                          transactions before freeze */
    uint32 pre_freeze_capture_time;   /* Capture time before freeze,
                                          usec unit */
    uint32 post_sample_probability;   /* sample probability at post
                                          stage, step is 1/16, valid
                                          configure range is 1 C 16, 1
                                          means 1/16 sample, 16 means
                                          sample all */
    uint32 pre_sample_probability;    /* sample probability at pre
```

```
                                        stage, step is 1/16, valid
                                        configure range is 1 C 16, 1
                                        means 1/16 sample, 16 means
                                        sample all */
} bcm_cosq_tcb_config_t;
```

Table: TCB monitor scope type

Name	Purpose
bcmCosqTcbScopeUcastQueue	Set scope to a unicast queue
bcmCosqTcbScopeIngressPort	Set scope to ingress port
bcmCosqTcbScopeEgressPort	Set scope to egress port

trigger_reason is used to configure the drop event trigger reason, 4 types are availible. Multiple reasons could be enabled at the same time.

```
/* TCB Track Trigger Reason */
#define BCM_COSQ_TCB_INGRESS_ADMIN_DROP    0x0001     /* Ingress
                                                      admission drop */
#define BCM_COSQ_TCB_EGRESS_ADMIN_DROP    0x0002     /* Egress
                                                      admission drop */
#define BCM_COSQ_TCB_WRED_DROP     0x0004     /* Weighted random
                                             drop */
#define BCM_COSQ_TCB_CFAP_DROP     0x0008     /* Cell free
                                             allocations pool drop */
```

RETURNS

None.

---

bcm_cosq_tcb_control_set

bcm_cosq_tcb_control_get

Set/get TCB working status.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tcb_control_set(int unit,
                             bcm_cosq_buffer_id_t buffer_id,
                             bcm_cosq_tcb_control_t type,
                             int arg);
int bcm_cosq_tcb_control_get(int unit,
                             bcm_cosq_buffer_id_t buffer_id,
                             bcm_cosq_tcb_control_t type,
                             int *arg);
```

PARAMETERS

unit	(IN) BCM device number
buffer_id	(IN) TCB buffer id
type	(IN) Pointer to profile index.
arg	(IN) (for "_set") Control argument.
arg	(OUT) (for "_get") Control argument.

DESCRIPTION

Available control type are Table: TCB Control Type When control type is bcmCosqTcbControlEnable, argument with value 1 means enable, 0 means disable. All TCB configuration can only be changed when TCB is disabled. When control type is bcmCosqTcbControlFreeze, argument with value 1 means freeze, 0 means restart.Buffer_id with an invalid value -1 could be used to freeze and restart all TCB buffers at a once.

Table: TCB Control Type

Type	Description
bcmCosqTcbControlEnable	Enable or disable TCB.
bcmCosqTcbControlFreeze	Freeze or restart TCB.

RETURNS

BCM_E_NONE

BCM_E_XXX

bcm_cosq_tcb_event_multi_get

Get drop event records in TCB buffer.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tcb_event_multi_get(int unit,
                                 bcm_cosq_buffer_id_t buffer_id,
                                 int array_max,
                                 bcm_cosq_tcb_event_t *event_array,
                                 int *array_count,
                                 int *overflow_count);
```

PARAMETERS

unit	(IN) BCM device number.
buffer_id	(IN) TCB buffer id.
array_max	(IN) Number of elements allocated in event_array.
event_array	(IN) Array of drop event records.
array_count	(IN) Actual number of packet records returned in event_array.
overflow_count	(IN) Count of drop events exceeding the event buffer storage.

DESCRIPTION

Get all drop event records when a packet drop occurs. This could only be used when TCB is in FREEZE state. When array_max is 0, event_array is null, array_count will return the actual record number in TCB buffer.

Drop event record format is as below:

```
typedef struct bcm_cosq_tcb_event_s {
    uint64 timestamp;           /* timestamp */
    bcm_gport_t ingress_gport;  /* ingress gport. */
```

```
    bcm_gport_t egress_gport;   /* egress gport. */
    bcm_gport_t queue;          /* Egress queue gport */
    uint32 queue_size;          /* Queue depth */
    uint32 service_pool_size;   /* Service pool depth */
    uint32 drop_reason;         /* Drop reason vector */
    bcm_color_t color;          /* Packet color */
} bcm_cosq_tcb_event_t;
```

RETURNS

BCM_E_NONE
BCM_E_XXX

bcm_cosq_tcb_gport_threshold_mapping_set

bcm_cosq_tcb_gport_threshold_mapping_get

Set/Get threshold profile for UCQ.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tcb_gport_threshold_mapping_set(int unit,
                                             bcm_cosq_object_id_t *id,
                                             int profile_index);
int bcm_cosq_tcb_gport_threshold_mapping_get(int unit,
                                             bcm_cosq_object_id_t *id,
                                             int *profile_index);
```

PARAMETERS

unit	(IN) BCM device number
id	(IN) Compound index containing gport, cosq, buffer. See (unknown XREF bcm_cosq_object_id_t_init) .
profile_index	(IN) (for "_set") Profile index.
profile_index	(OUT) (for "_get") Pointer to profile index.

DESCRIPTION

Mapping one UCQ or multiple UCQs to a threshold profile. id.port can be a local physical port or a UCQ gport. When id.port is a physical port, bcm_cosq_tcb_gport_threshold_mapping_set would attach all UCQs in the specified port to the map, bcm_cosq_tcb_gport_threshold_mapping_get would get profile index of the first UCQ in this port .

RETURNS

BCM_E_NONE
BCM_E_XXX

bcm_cosq_tcb_threshold_profile_create

Create tcb threshold profile.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tcb_threshold_profile_create(int unit,
                                          uint32 flags,
                                          bcm_cosq_buffer_id_t buffer_id,
                                          bcm_cosq_tcb_threshold_profile_t
*threshold,
                                          int *profile_index);
```

PARAMETERS

unit	(IN) BCM device number
flags	(IN) TCB threshold profile create flags.
buffer_id	(IN) TCB buffer id
threshold	(IN) TCB configuration
profile_index	(IN/OUT) Pointer to profile index.

DESCRIPTION

Create a threshold profile for a specified TCB. The available flags for creation are Table: TCB Profile Create Flags The profile_index could be created and returned by API. User should specify a profile_index when BCM_COSQ_TCB_THRESHOLD_PROFILE_WITH_ID flag is set. WhenBCM_COSQ_TCB_THRESHOLD_PROFILE_UPDATE is set, BCM_COSQ_TCB_THRESHOLD_PROFILE_WITH_ID should also be set, the threshole profile with specified profile_index will be updated. For threshold profile, start threshold should bigger than stop threshold.

Table: TCB Profile Create Flags

Flag	Description
BCM_COSQ_TCB_THRESHOLD_PROFILE_WITH_ID	Create a threshold profile with a specified id.
BCM_COSQ_TCB_THRESHOLD_PROFILE_UPDATE	Update an existing threshold profile.

bcm_cosq_tcb_threshold_profile_destroy

Destroy a tcb threshold profile.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tcb_threshold_profile_destroy(int unit,
                                           bcm_cosq_buffer_id_t buffer_id,
                                           int profile_index);
```

PARAMETERS

unit	(IN) BCM device number
buffer_id	(IN) TCB buffer id
profile_index	(IN) Pointer to profile index.

DESCRIPTION

Destory a threshold profile for a specified TCB.

RETURNS

BCM_E_NONE
BCM_E_XXX

bcm_cosq_tcb_threshold_profile_get

Get a tcb threshold profile.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tcb_threshold_profile_get(int unit,
                                       bcm_cosq_buffer_id_t buffer_id,
                                       int profile_index,
                                       bcm_cosq_tcb_threshold_profile_t
*threshold);
```

PARAMETERS

unit	(IN) BCM device number
buffer_id	(IN) TCB buffer id
profile_index	(IN) Pointer to profile index.
threshold	(OUT) TCB configuration

DESCRIPTION

Get a threshold profile for a specified TCB.

RETURNS

BCM_E_NONE
BCM_E_XXX

bcm_cosq_tcb_threshold_profile_t_init

Initialize a tcb threshold profile structure.

SYNOPSIS

```
#include <bcm/cosq.h>
void
bcm_cosq_tcb_threshold_profile_t_init(bcm_cosq_tcb_threshold_profile_t_init
*threshold);
```

PARAMETERS

threshold	Pointer to TCB threshold profile structure to initialize.

DESCRIPTION

Initialize a tcb threshold profile structure.

```
typedef struct bcm_cosq_tcb_threshold_profile_s {
    uint32 start_threshold_bytes;   /* The start threshold above which
                                       queue start the capture. */
    uint32 stop_threshold_bytes;    /* The stop threshold below which
                                       queue stop the capture. */
} bcm_cosq_tcb_threshold_profile_t;
```

RETURNS

BCM_E_NONE
BCM_E_XXX

Section 6.2: Explicit Congestion Notification (ECN)

Table: ECN Map Flags

Flag	Description	
BCM_ECN_EGRESS_PORT_INT_CN_TO_EXP_MAP	If set, indicate to set/get INT_CN_TO_EXP mapping on egress port.	New
BCM_ECN_EGRESS_PORT_ECN_TO_EXP_MAP	If set, indicate to set/get ECN_TO_EXP mapping on egress port.	New
BCM_ECN_INGRESS_PORT_TUNNEL_TERM_MAP	If set, indicate to set/get tunnel termination mapping on egress port.	New

Table: ECN Traffic Action Flags

Flag	Description	
BCM_ECN_TRAFFIC_ACTION_INGRESS_ECN_MARKING	If set, indicates that the packet's ECN bits need to be updated.	New
BCM_ECN_TRAFFIC_ACTION_INGRESS_DROP	If set, indicates to drop packet in ingress stage.	New
BCM_ECN_TRAFFIC_ACTION_RESPONSIVE	If set, indicates that the configuration of this mapping is for ecn responsive traffic.	New
BCM_ECN_TRAFFIC_ACTION_EGRESS_EXP_MARKING	If set, indicates that the packet's EXP bits need to be updated.	New

Table: ECN Map Flags

Flag	Description	
BCM_ECN_MAP_TUNNEL_TERM	If set, indicate to create a tunnel termination mapping profile.	New
BCM_ECN_MAP_MPLS_INT_CN_TO_EXP	If set, indicate to create an INT_CN_TO_EXP mapping profile. Can be only used with BCM_ECN_MAP_EGRESS.	New
BCM_ECN_MAP_EGRESS	If set, indicate to create an egress map used in conjunction with map type.	New

BCM_ECN_MAP_INGRESS	If set, indicate to create an ingress map used in conjunction with map type.	New
BCM_ECN_MAP_MPLS	If set, indicate to create an MPLS type of map, used with BCM_ECN_MAP_INGRESS or BCM_ECN_MAP_EGRESS. When it is used with BCM_ECN_MAP_INGRESS, it means EXP_TO_IP_ECN mapping, and when it is used with BCM_ECN_MAP_EGRESS , it means IP_ECN_TO_EXP mapping.	New
BCM_ECN_MAP_WITH_ID	If set, the ECN map ID is specified by users.	New

Table: ECN Traffic Map Flags

Flag	Description	
BCM_ECN_TRAFFIC_MAP_RESPONSIVE	If set, indicates the value of responsive indication is one. Otherwise, responsive indication is zero.	New

New APIs for Section 6.2: Explicit Congestion Notification (ECN)

bcm_ecn_map_create

To create an ECN mapping profile.

SYNOPSIS

```
#include <bcm/ecn.h>
bcm_ecn_map_create(int unit, uint32 flags, int *ecn_map_id)
```

PARAMETERS

unit	BCM unit number.
flags	BCM ecn map flags.
ecn_map_id	BCM ecn map id.

DESCRIPTION

To create an ECN mapping profile based on flags.

Flags are listed in the following tables. Table: ECN Map Flags

Table: ECN Map Flags

Flag	Description
BCM_ECN_MAP_WITH_ID	If set, the ECN map ID is specified by users.
BCM_ECN_MAP_INGRESS	If set, indicate to create an ingress map used in conjunction with map type.
BCM_ECN_MAP_EGRESS	If set, indicate to create an egress map used in conjunction with map type.
BCM_ECN_MAP_MPLS	If set, indicate to create an MPLS type of map, used with BCM_ECN_MAP_INGRESS or BCM_ECN_MAP_EGRESS. When it is used with

	BCM_ECN_MAP_INGRESS, it means EXP_TO_IP_ECN mapping, and when it is used with BCM_ECN_MAP_EGRESS , it means IP_ECN_TO_EXP mapping.
BCM_ECN_MAP_MPLS_INT_CN_TO_EXP	If set, indicate to create an INT_CN_TO_EXP mapping profile. Can be only used with BCM_ECN_MAP_EGRESS.
BCM_ECN_MAP_TUNNEL_TERM	If set, indicate to create a tunnel termination mapping profile.

RETURNS

BCM_E_NONE	Success
BCM_E_XXX	Other error

bcm_ecn_map_destroy

To destroy an ECN mapping profile.

SYNOPSIS

```
#include <bcm/ecn.h>
bcm_ecn_map_destroy(int unit,  int ecn_map_id);
```

PARAMETERS

unit	BCM unit number.
ecn_map_id	BCM ecn map id.

DESCRIPTION

To destroy an ECN mapping profile.

RETURNS

BCM_E_NONE	Success
BCM_E_XXX	Other error

bcm_ecn_map_get

To get an ECN mapping entry from an ECN mapping profile.

SYNOPSIS

```
#include <bcm/ecn.h>
bcm_ecn_map_get(int unit, int ecn_map_id , bcm_ecn_map_t *ecn_map)
```

PARAMETERS

unit	BCM unit number.
ecn_map_id	BCM ecn map id.
ecn_map	BCM ecn map structure.

DESCRIPTION

To get an ECN mapping entry from an ECN mapping profile.

RETURNS

BCM_E_NONE	Success
BCM_E_XXX	Other error

bcm_ecn_map_set

To set an ECN mapping entry in an ECN mapping profile.

SYNOPSIS

```
#include <bcm/ecn.h>
bcm_ecn_map_set(int unit, uint32 options, int ecn_map_id , bcm_ecn_map_t
*map)
```

PARAMETERS

unit	BCM unit number.
options	BCM ecn options.
ecn_map_id	BCM ecn map id.
map	BCM ecn map structure pionter.

DESCRIPTION

To set an ECN mapping entry in the specfic profile.

Ecn map configuration is listed in the following tables. Table: ECN Map structure This ecn map structure can be used to set entries in four types of ecn mappings. MPLS EXP to IP ECN mapping (used in L2/L3 VPN LER termination and LSR ingress stage) ecn and exp value inbcm_ecn_map_t are used to get a new ecn.
BCM_ECN_TRAFFIC_ACTION_INGRESS_DROP can be used in bcm_mpls_ecn_map_t.action_flags. IP ECN to EXP mapping (used in LER initiation) ecn and exp value in bcm_ecn_map_t are used to get a new exp.
BCM_ECN_TRAFFIC_ACTION_EGRESS_DROP andBCM_ECN_TRAFFIC_ACTION_EGRESS_EXP_MARKING can be used in bcm_ecn_map_t.action_flags. INT CN to EXP mapping (used in LSR egress stage) int_cn and exp value in bcm_mpls_ecn_map_t are used to get a new exp.
BCM_ECN_TRAFFIC_ACTION_EGRESS_DROP andBCM_ECN_TRAFFIC_ACTION_EGRESS_EXP_MARKING can be used in bcm_ecn_map_t.action_flags. Tunnel termination ECN mapping inner_ecn and ecn in bcm_mpls_ecn_map_t are used to get a new ecn.
BCM_ECN_TRAFFIC_ACTION_INGRESS_ECN_MARKING and BCM_ECN_TRAFFIC_ACTION_INGRESS_DROP can be used inbcm_ecn_map_t.action_flags.

Table: ECN Map structure

Parameter	Description
action_flags	action flags for ECN.@xref=BCM_ECN_MAP_ACTION_FLAGS_f
int_cn	internal cn as key.

inner_ecn	inner packet ip ecn as key
ecn	outer packet ip ecn as key
exp	mpls_exp as key.
new_ecn	new ecn value should be updated in ip packet.
new_exp	new mpls_exp should be updated in MPLS packet.

RETURNS

BCM_E_NONE	Success
BCM_E_XXX	Other error

bcm_ecn_map_t_init

To init an ECN mapping structure.

SYNOPSIS

```
#include <bcm/ecn.h>
bcm_ecn_map_t_init(bcm_ecn_map_t *ecn_map);
```

PARAMETERS

ecn_map	BCM ecn map structure.

DESCRIPTION

To initialize an ECN mapping structure.

RETURNS

Nothing

bcm_ecn_port_map_set

bcm_ecn_port_map_get

To set/get the egress ecn_map_id on the specfic port.

SYNOPSIS

```
#include <bcm/ecn.h>
bcm_ecn_port_map_set(int unit, bcm_gport_t port,  bcm_ecn_port_map_t
*ecn_port_map)
bcm_ecn_port_map_get(int unit, bcm_gport_t port,  bcm_ecn_port_map_t
*ecn_port_map)
```

PARAMETERS

unit	BCM unit number.
port	BCM gport.
ecn_port_map	BCM ecn map port structure.

DESCRIPTION

To set the egress ecn_map_id on the specfic port.

Ecn map port structure is listed in the following tables. Table: ECN Port Map structure

Table: ECN Port Map structure

Parameters	Description
flags	used to specify which mapping to be set/get on port.
ecn_map_id	could be INT_CN_TO_EXP mapping or IP_ECN_TO_EXP mapping id.

Flags are listed in the following tables. Table: ECN Map Flags

Table: ECN Map Flags

Flag	Description
BCM_ECN_EGRESS_PORT_ECN_TO_EXP_MAP	If set, indicate to set/get ECN_TO_EXP mapping on egress port.
BCM_ECN_EGRESS_PORT_INT_CN_TO_EXP_MAP	If set, indicate to set/get INT_CN_TO_EXP mapping on egress port.
BCM_ECN_INGRESS_PORT_TUNNEL_TERM_MAP	If set, indicate to set/get tunnel termination mapping on egress port.

RETURNS

BCM_E_NONE	Success
BCM_E_XXX	Other error

bcm_ecn_port_map_t_init

To initialize an ecn port map structure.

SYNOPSIS

```
#include <bcm/ecn.h>
bcm_ecn_port_map_t_init(bcm_ecn_port_map_t *ecn_port_map)
```

PARAMETERS

ecn_port_map	BCM ecn map port structure.

DESCRIPTION

To initialize an ecn port map structure.

RETURNS

Nothing

Section 6.3: Port Extension Management

Table: Extender egress object flags

Name	Purpose	
BCM_EXTENDER_EGRESS_SERVICE_PRI_REPLACE	Replace SD-tag Priority	New
BCM_EXTENDER_EGRESS_SERVICE_VLAN_REPLACE	Replace SD-tag VLAN ID	New
BCM_EXTENDER_EGRESS_SERVICE_VLAN_DELETE	Delete SD-tag	New
BCM_EXTENDER_EGRESS_SERVICE_VLAN_ADD	Add SD-tag	New
BCM_EXTENDER_EGRESS_L3	Create L3 egress object	New
BCM_EXTENDER_EGRESS_MULTICAST	Create a multicast Extender egress object	New
BCM_EXTENDER_EGRESS_SERVICE_TPID_REPLACE	Replace SD-tag TPID	New

New structure

```
typedef struct bcm_extender_egress_s {
    uint32 flags;                       /* BCM_Extender_EGRESS_xxx */
    bcm_gport_t port;                   /* Physical port / trunk */
    uint16 extended_port_vid;           /* Extender port VID. */
    bcm_vlan_t match_vlan;              /* Outer VLAN ID to match. */
    bcm_extender_pcp_de_select_t pcp_de_select; /* Selection of PCP and DE
fields for
                                           egress ETAG. */
    int qos_map_id;                     /* Qos map id for egress etag
mapping profile. */
    uint8 pcp;                          /* Default PCP field of ETAG. */
```

```
    uint8 de;                           /* Default DE field of ETAG. */
    bcm_if_t egress_if;                 /* Egress object ID */
    int service_qos_map_id;             /* Egress SD-tag Priority and CFI
mapping
                                           profile ID */
    uint16 service_tpid;                /* Egress SD-tag TPID */
    bcm_vlan_t service_vlan;            /* Egress SD-tag VLAN ID */
    uint8 service_pri;                  /* Egress SD-tag default Priority
*/
    uint8 service_cfi;                  /* Egress SD-tag default CFI */
    bcm_if_t intf;                      /* L3 interface */
    uint32 multicast_flags;             /* BCM_L3_MULTICAST flag
definitions */
} bcm_extender_egress_t;
```

New APIs for Section 6.3: Port Extension Management

bcm_extender_egress_add

Create a Extender egress object and add it to a Extender port.

SYNOPSIS

```
#include <bcm/extender.h>
int bcm_extender_egress_add(int unit, bcm_gport_t extender_port,
bcm_extender_egress_t *extender_egress);
```

PARAMETERS

unit	(IN) BCM device number
extender_port	(IN) Extender GPORT ID
extender_egress	(IN/OUT) Extender egress object, the egress_if field is an output.

DESCRIPTION

Create a Extender egress object and add it to a Extender port. The Extender egress object's ID is returned in the extender_egress structure's egress_if field.

RETURNS

BCM_E_XXX

bcm_extender_egress_delete

Delete a Extender egress object from a Extender port and destroy it.

SYNOPSIS

```
#include <bcm/extender.h>
int bcm_extender_egress_delete(int unit, bcm_gport_t extender_port,
bcm_extender_egress_t *extender_egress);
```

PARAMETERS

unit	(IN) BCM device number
extender_port	(IN) Extender GPORT ID
extender_egress	(IN) Extender egress object

DESCRIPTION

Delete a Extender egress object from a Extender port. The Extender egress object is then destroyed. The Extender egress object is identified by the extender_egress structure's port, extended_port_vid, and optional match_vlan fields, not by the egress_if field.

RETURNS

BCM_E_XXX

bcm_extender_egress_delete_all

Delete all Extender egress objects associated with a Extender port.

SYNOPSIS

```
#include <bcm/extender.h>
int bcm_extender_egress_delete_all(int unit, bcm_gport_t extender_port);
```

PARAMETERS

unit	(IN) BCM device number
extender_port	(IN) Extender GPORT ID

DESCRIPTION

Delete all Extender egress objects associated with a Extender port. These Extender egress objects are then destroyed.

RETURNS

BCM_E_XXX

bcm_extender_egress_get

Get Extender egress objects associated with a Extender port.

SYNOPSIS

```
#include <bcm/extender.h>
int bcm_extender_egress_get(int unit, bcm_gport_t extender_port,
                        bcm_extender_egress_t *extender_egress);
```

PARAMETERS

unit	(IN) BCM device number

extender_port	(IN) Extender GPORT ID
extender_egress	(IN/OUT) Array of Extender egress objects

DESCRIPTION

With Extender port, port and extended_port_vid input, a specific egress object can be retrieved.

RETURNS

BCM_E_XXX

bcm_extender_egress_get_all

Get Extender egress objects associated with a Extender port.

SYNOPSIS

```
#include <bcm/extender.h>
int bcm_extender_egress_get_all(int unit, bcm_gport_t extender_port, int
array_size,
                        bcm_extender_egress_t *extender_egress_array, int
*count);
```

PARAMETERS

unit	(IN) BCM device number
extender_port	(IN) Extender GPORT ID
array_size	(IN) Number of Extender egress objects allocated in extender_egress_array
extender_egress_array	(IN/OUT) Array of Extender egress objects
count	(OUT) Number of Extender egress objects returned

DESCRIPTION

Retrieve Extender egress objects associated with a Extender port. If array_size = 0 and extender_egress_array = NULL, the number of Extender egress objects associated with the Extender port is returned in the parameter count.

RETURNS

BCM_E_XXX

bcm_extender_egress_set

Create a set of Extender egress objects and add them to a Extender port.

SYNOPSIS

```
#include <bcm/extender.h>
int bcm_extender_egress_set(int unit, bcm_gport_t extender_port, int
array_size,
                        bcm_extender_egress_t *extender_egress_array);
```

PARAMETERS

unit	(IN) BCM device number
extender_port	(IN) Extender GPORT ID
array_size	(IN) Number of Extender egress objects in extender_egress_array
extender_egress_array	(IN/OUT) Array of Extender egress objects

DESCRIPTION

Delete existing Extender egress objects associated with the Extender port. Then, create a set of new Extender egress objects and add them to the Extender port. In the extender_egress_array, the ID of each Extender egress object is returned in the egress_if field.

RETURNS

BCM_E_XXX

bcm_extender_egress_t_init

Initialize the Extender egress object structure.

SYNOPSIS

```
#include <bcm/extender.h>
void bcm_extender_egress_t_init(bcm_extender_egress_t *extender_egress);
```

PARAMETERS

extender_egress	(IN/OUT) Pointer to the struct to be initialized

DESCRIPTION

Initialize the Extender egress object structure.

RETURNS

None.

Section 6.4: Fabric

New APIs for Section 6.4: Fabric

bcm_fabric_static_replication_set

Set the replications in the fabric element to external faps

SYNOPSIS

```
#include <bcm/fabric.h>
int
bcm_fabric_static_replication_set(
    int unit,
    bcm_port_t port,
    uint32 flags,
    uint32 destid_count,
    bcm_module_t *destid_array);
```

PARAMETERS

unit	(IN) Unit number.
port	(IN) Port number.
flags	(IN) Configuration parameters.
destid_count	(IN) Size of destid_array.
destid_array	(IN) Array of destination IDs.

DESCRIPTION

destid_array represent either modid or group of modids.

RETURNS

BCM_E_xxx

Section 6.5: FCMAP API

New MACRO

```
#define BCM_FCMAP_ATTR2_TRCM_ATTRIBS_MASK   (0x1 << 21)
#define BCM_FCMAP_ATTR2_COS_TO_PRI_MASK     (0x1 << 22)


#define BCM_FCMAP_PORT_MODULE_NO_FW_I2C          (0x1 << 0)
#define BCM_FCMAP_PORT_MODULE_IS_COPPER          (0x1 << 1)
#define BCM_FCMAP_PORT_MODULE_SPEED_CAP32G       (0x1 << 2)
#define BCM_FCMAP_PORT_MODULE_SPEED_CAP16G       (0x1 << 3)
#define BCM_FCMAP_PORT_MODULE_SPEED_CAP8G        (0x1 << 4)
#define BCM_FCMAP_PORT_MODULE_SPEED_CAP4G        (0x1 << 5)
#define BCM_FCMAP_PORT_MODULE_FEC_CAP32G         (0x1 << 6)
#define BCM_FCMAP_PORT_MODULE_FEC_CAP16G         (0x1 << 7)
#define BCM_FCMAP_PORT_MODULE_IS_QSFP            (0x1 << 8)
#define BCM_FCMAP_PORT_MODULE_SUPP_RS            (0x1 << 9)
#define BCM_FCMAP_PORT_MODULE_RS_HRD_RX_CUTOFF   (0x3 << 10)
#define BCM_FCMAP_PORT_MODULE_RS_HRD_TX_CUTOFF   (0x3 << 12)
```

Section 6.6: Field Processor

Table: Field Qualify Presel Staggred

FIELD_QUALIFY_PRESEL_STAGGERED	OR with ID when calling bcm_field_qualify_* functions to indicate ID is a preselector in staggered mode	New

Table: Packet Resolution Values (for bcm_field_qualify_PacketRes)

BCM_FIELD_PKT_RES_xxx	Purpose	
BCM_FIELD_PKT_RES_PIMBIDIRUNKNOWN	Unknown PIM-BIDIR packet from DF	New

Table: Header format types

Header format type	Description	
bcmFieldHeaderFormatCount	Always Last. Not a usable value.	Modified

Table: Field Qualifiers

bcmFieldQualifyStaggeredValue0	To qualify on value taken from action result 0 in staggered mode.
bcmFieldQualifyStaggeredValue1	To qualify onvalue taken from action result 1 in staggered mode.
bcmFieldQualifyStaggeredValue2	To qualify on value taken from action result 2 in staggered mode.
bcmFieldQualifyStaggeredValue3	To qualify onvalue taken from action result 3 in staggered mode.
bcmFieldQualifyStaggeredDirectValue	*To qualify on value taken from KAPS payload in staggered mode.
bcmFieldQualifyStaggeredPreselProfile0	program selection profile value taken from action result 0 in staggered mode.
bcmFieldQualifyStaggeredPreselProfile1	program selection profile value taken from action result 1 in staggered mode.

bcmFieldQualifyStaggeredPreselProfile2	program selection profile value taken from action result 2 in staggered mode.
bcmFieldQualifyStaggeredPreselProfile3	program selection profile value taken from action result 3 in staggered mode.
bcmFieldQualifyStaggeredPreselProfileDirect	program selection profile value taken from KAPS payload 0 in staggered mode.
bcmFieldQaulifyPreselId	Program selection value in 2nd Pass according to the 1st Pass preselector-id.

New Field Actions for bcm_field_action_add

Table: Field Actions for bcm_field_action_add

bcmFieldActionEgressTimeStampInsert	Insert EPIPE timestamp.	n/a	n/a
bcmFieldActionIngressTimeStampInsert	Insert IPIPE timestamp.	n/a	n/a
bcmFieldActionIngressTimeStampInsertCancel	Do not insert ingress timestamp. Overrules port config.	n/a	n/a
bcmFieldActionEgressTimeStampInsertCancel	Do not insert egress timestamp. Overrules port config.	n/a	n/a
bcmFieldActionMplsLabel1ExpNew	Change the EXP field in forwarding label(label one) of MPLS packets.	New EXP value	n/a
bcmFieldActionDynamicEcmpEnable	Force packets to use DLB when matched by IFP rules, even though the Group Table indicates hash-based load balancing is to be used.	n/a	n/a
bcmFieldActionDynamicTrunkAllEnable	Force packets to use DLB when matched by IFP rules, even though the Group Table indicates hash-based load balancing is to be used.	n/a	n/a
bcmFieldActionDynamicEcmpCancel	Same as bcmFieldActionEcmpLoadBalanceCancel.	n/a	n/a
bcmFieldActionDynamicTrunkAllCancel	Force packets to use hash-based load balancing, even though the group is configured to support DLB.	n/a	n/a
bcmFieldActionDgm	Indicate IFP action is valid for DGM. It is mandatory to add this action to for the following three actions to get applied.	n/a	n/a

bcmFieldActionDgmThreshold	Indicates the IFP action is only valid when the primary optimal band is equal to or below this threshold. If threshold is set to 7, this action will always be valid.	Threshold value in DGM, range 0-7	n/a
bcmFieldActionDgmBias	Indicates the bias in favor of the alternate path in DGM.	Bias value in DGM, range 0-7	n/a
bcmFieldActionDgmCost	Indicates the cost of switching over to the alternate path in DGM.	Cost value in DGM, range 0-7	n/a
bcmFieldActionEnableVlanCheck	Check VLAN membership and STG state. If this action is applied for MPLS Flows(VPLS, VPWS) in VFP, resolution of Source and Destination ports will be affected because for these flows resolution of Source and destination port happens via SVP/DVPs and physical parameters like port, vlan should not be used in such cases	n/a	n/a
bcmFieldActionAdmitProfile	Return admit profile by bcmFieldAdmitProfile_t	n/a	n/a

New Field Action flags

```
/*
 * Bitmap for the various options on 'flags' on
 * bcm_field_action_width_set() or bcm_field_action_width_get()
 */
typedef enum bcm_field_action_width_e {
    bcm_default_action_width = 0    /* When this bit is set then ignore
'width'
                                       and set/get action runtime width to
its
                                       default value. */
} bcm_field_action_width_t;
```

Modified bcm_field_stage_info

```
typedef struct bcm_field_stage_info_s {
    int field_presel_advanced_mode;     /* is presel advanced mode enabled.
```

```
*/
    int field_presel_max_id;            /* The maximum presel id that can
be
                                           defined for a new created
program
                                           selector in advanced mode. */
    int field_presel_staggered_max_id;  /* The maximum presel id that can
be
                                           defined for a new created
program
                                           selector in staggered mode. */
 } bcm_field_stage_info_t;


/*
 * Used for action bcmFieldActionAdmitProfile to return profile number by
 * enum.
 */
typedef enum bcm_field_admit_profile_e {
    bcmFieldAdmitProfileEcnNonCapable = 0, /* Packet is not ECN capable.
Use
                                           profile 0. */
    bcmFieldAdmitProfileEcnCapable = 1  /* Packet is ECN capable. Use
profile 1. */
} bcm_field_admit_profile_t;
```

New APIs for Section 6.6: Field Processor

bcm_field_class_map_get

To get the set of mapped TTL/ToS/TCP values of the packet for a given set of TTL/ToS/TCP values respectively.

SYNOPSIS

```
#include <bcm/field.h>
bcm_error_t
bcm_field_class_map_get (int unit,
        bcm_field_stage_t stage,
        bcm_field_qualify_t qual,
        uint32 num_entries,
```

```
        uint32 *orig_values_array,
        uint32 *map_values_array);
```

PARAMETERS

unit	(IN) Unit number
stage	(IN) Field Stage of type bcm_field_stage_t
qual	(IN) Field qualifier of type bcm_field_qualify_t
num_entries	(IN) size of the Orig_values array
orig_values_array	(IN) List of TTL/TOS/TCPFlags Values
map_values_array	(OUT) List of Mapped Values for each TTL/TOS/TCPFlags

DESCRIPTION

To get the set of mapped TTL/ToS/TCP values of the packet for a given set of TTL/ToS/TCP values respectively. Note that this API is supported only in Ingress based Field Processor(IFP) Stage.

RETURNS

bcm_error_t

---

bcm_field_class_map_set

To map a given set of TTL/ToS/TCP values of the packet to another set of TTL/ToS/TCP values respectively.

SYNOPSIS

```
#include <bcm/field.h>
bcm_error_t
bcm_field_class_map_set (int unit,
        bcm_field_stage_t stage,
        bcm_field_qualify_t qual,
        uint32 num_entries,
        uint32 *orig_values_array,
        uint32 *map_values_array);
```

PARAMETERS

unit	(IN) Unit number
stage	(IN) Field Stage of type bcm_field_stage_t
qual	(IN) Field qualifier of type bcm_field_qualify_t
num_entries	(IN) size of the Orig_values array
orig_values_array	(IN) List of TTL/TOS/TCPFlags Values
map_values_array	(IN) List of Mapped Values for each TTL/TOS/TCPFlags

DESCRIPTION

To map a given set of TTL/ToS/TCP values of the packet to another set of TTL/ToS/TCP values respectively. The field qualifiers bcmFieldQualifyTtl, bcmFieldQualifyTos, bcmFieldQualifyTcpControl and their respective ClassZero qualifiers bcmFieldQualifyTtlClassZero , bcmFieldQualifyTosClassZero , bcmFieldQualifyTcpClassZero can be interchangeably used. If both ClassZero and ClassOne of same type (TTL, TOS, or TCPFlags) are part of the group qset then the group will be created in slice-pairing double wide mode. If all 3 qualifiers of same type (TTL/TOS/TCPFlags) are part of group qset then group creation fails. For ex: bcmFieldQualifyTtl, bcmFieldQualifyTtlClassZero and bcmFieldQualifyTtlClassOne.

Note that this API is supported only in Ingress based Field Processor(IFP) Stage.

API programming Recommendation: Users are free to change mapped values before/after IFP rules are created to match the mapped values. There will be lesser chances of programming errors if mapped values are set before FP rules are created. If user wants to change the mapped values after FP rules are created, do it carefully to avoid FALSE FP rule hit/miss. Recommendation is to avoid changing the mapped values after FP rules are created.

RETURNS

bcm_error_t

bcm_field_group_ports_get

Retrieve the bitmap of ports associated with a given field group.

SYNOPSIS

```
#include <bcm/field.h>
int
bcm_field_group_ports_get(
    int unit,
    bcm_field_group_t group,
    bcm_pbmp_t *pbmp);
```

PARAMETERS

unit	(IN) Unit number
group	(IN) Field group ID
pbmp	(OUT) Field group pbmp

DESCRIPTION

Retrieve the bitmap of ports associated with a given field group ID.

RETURNS

int

---

bcm_field_qualify_PreselId

Set match criteria for bcmFieldQaulifyPreselId.

SYNOPSIS

```
#include <bcm/field.h>
int
bcm_field_qualify_PreselId (
    int unit,
    bcm_field_entry_t entry,
    uint32 data,
    uint32 mask);
```

PARAMETERS

unit	(IN) Unit number
entry	(IN) Field entry ID
data	(IN) Qualifier match data
mask	(IN) Qualifier match mask

DESCRIPTION

Set match criteria for bcmFieldQaulifyPreselId.

RETURNS

int

---

bcm_field_qualify_PreselId_get

Get match criteria for bcmFieldQaulifyPreselId.

SYNOPSIS

```
#include <bcm/field.h>
int
bcm_field_qualify_PreselId_get (
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);
```

PARAMETERS

unit	(IN) Unit number
entry	(IN) Field entry ID
data	(OUT) Qualifier match data
mask	(OUT) Qualifier match mask

DESCRIPTION

Get match criteria for bcmFieldQaulifyPreselId.

RETURNS

int

Section 6.7: Kernel Network (KNET) Configuration

Parameter bcm_cos_queue_t cosq added to bcm_knet_netif_t

```
typedef struct bcm_knet_netif_s {
    int id;                                 /* Network interface ID. */
    int type;                               /* Network interface type
                                               (BCM_KNET_NETIF_T_XXX). */
    uint32 flags;                           /* BCM_KNET_NETIF_F_XXX flags. */
    bcm_mac_t mac_addr;                     /* MAC address associated with this
                                               network interface. */
    bcm_vlan_t vlan;                        /* VLAN ID associated with this
                                               interface. */
    bcm_port_t port;                        /* Local port associated with this
                                               interface. */
    bcm_cos_queue_t cosq;                   /* Cos offset from base queue of
port
                                               associated with this interface.
*/
    char name[BCM_KNET_NETIF_NAME_MAX]; /* Network interface name (assigned
by
                                               kernel) */
} bcm_knet_netif_t;
```

Section 6.8: Layer 2 Address Management

Table: BCM Layer 2 Flags

Name	Purpose	
BCM_L2_SET_ENCAP_INVALID	DNX only: encap_id is added to MAC table as is, however valid bit is cleared.	New

Section 6.9: Layer 3 Management

Table: BCM L3 ECMP Member Flags

Name	Purpose	
BCM_L3_ECMP_MEMBER_DGM_ALTERNATE	If set indicating ECMP member is within alternate path, else within primary path.	New

Table: BCM L3 ECMP Dynamic Load Balancing Mode Flags

Name	Purpose	
BCM_L3_ECMP_DYNAMIC_MODE_DGM	ECMP dynamic group multipath mode: if primary path is good enough, follow primary path, else follow alternate path.	New

Table: BCM Layer3 Ingress Interface Flags

Name	Purpose	
BCM_L3_INGRESS_TUNNEL_TERM_ECN_MAP	To set the tunnel termination ecn map id.	New

Modified

```
/*
 * L3 Egress Structure.
 *
 * Description of an L3 forwarding destination.
 */
typedef struct bcm_l3_egress_s {
...
+    int dynamic_queue_size_weight;      /* Weight of queue size in
determining a
+                                          dynamic load balancing member's
```

```
+                                              quality. */
...
+    int mpls_ecn_map_id;                /* IP ECN/INT CN to MPLS EXP map
ID. */
} bcm_l3_egress_t;
```

New

```
/* L3 DGM structure */
typedef struct bcm_l3_dgm_s {
    uint32 threshold;   /* Indicates the primary path is always selected
when
                           primary optimal quality band is above this
threshold. */
    uint32 cost;        /* Indicates the quality band cost of switching
over to
                           alternate path. */
    uint32 bias;        /* Indicates the quality band bias in favor of
alternate
                           path. */
} bcm_l3_dgm_t;
```

New

```
typedef struct bcm_l3_egress_ecmp_s {
    uint32 flags;                       /* See BCM_L3_xxx flag definitions.
*/
    bcm_if_t ecmp_intf;                 /* L3 interface ID pointing to
egress
                                           ecmp object. */
    int max_paths;                      /* Max number of paths in ECMP
group. If
                                           max_paths <= 0, the default max
path
                                           which can be set by the API
                                           bcm_l3_route_max_ecmp_set will
be
                                           picked. */
    uint32 ecmp_group_flags;            /* BCM_L3_ECMP_xxx flags. */
    uint32 dynamic_mode;                /* Dynamic load balancing or
resilient hashing
                                           mode. See
BCM_L3_ECMP_DYNAMIC_MODE_xxx
                                           definitions. */
```

```
    uint32 dynamic_size;                /* Number of flows for dynamic load
                                           balancing (DLB) or resilient
hashing (RH).
                                           Valid DLB values are 512, 1k,
doubling up to 32k.
                                           Valid RH values are 64, 128,
doubling up to 64k. */
    uint32 dynamic_age;                 /* Inactivity duration, in
microseconds. */
    uint32 dynamic_load_exponent;       /* The exponent used in the
                                           exponentially weighted moving
average
                                           calculation of historical member
                                           load. */
    uint32 dynamic_expected_load_exponent; /* The exponent used in the
                                           exponentially weighted moving
average
                                           calculation of historical
expected
                                           member load. */
+    bcm_l3_dgm_t dgm;                      /* DGM properties */
} bcm_l3_egress_ecmp_t;
```

Section 6.10: Link Monitoring and Notification

New APIs for Section 6.10: Link Monitoring and Notification

bcm_linkscan_trigger_event_set

bcm_linkscan_trigger_event_get

Get/set the link event from/to Linkscan function.

SYNOPSIS

```
#include <bcm/link.h>
int bcm_linkscan_trigger_event_set(int unit, bcm_port_t port, uint32 flags,
bcm_linkscan_trigger_event_t trigger_event, int enable);
int bcm_linkscan_trigger_event_get(int unit, bcm_port_t port, uint32 flags,
bcm_linkscan_trigger_event_t trigger_event, int *enable);
```

PARAMETERS

unit	BCM device number.
port	Device port number, the only acceptable value is -1.
flags	Flags(unused).
trigger_event	Link event type, see table (unknown XREF BCM_LINKSCAN_TRIGGER_EVENT_e.)
enable	Enable the link event.

DESCRIPTION

Get or set the link event from or to Linkscan function. The possible link event is described in table (unknown XREF BCM_LINKSCAN_TRIGGER_EVENT_e.) The link event can only be enabled and cannot be disabled.

The parameter 'port' can only be set to -1, which stands for ALL PORTS.

Table: Link event types

Type	Meaning
BCM_LINKSCAN_TRIGGER_EVENT_FAULT	Enable link fault checking in Linkscan thread

RETURNS

BCM_E_XXX

Section 6.11: Mirroring

New Macro BCM_MIRROR_DEST_EGRESS_TRAP_WITH_SYSTEM_HEADER

For packets recycled by egress applications and not forwarded, retain the incoming packet's system headers as received from the fabric, and in case of an Egress MC packet, stamp the FTMH.OutLIF field with the CUD.

```
#define BCM_MIRROR_DEST_EGRESS_TRAP_WITH_SYSTEM_HEADER (1 << 28)  /*  */
```

Section 6.12: Packet Transmit and Receive

Added to Rx Decap Tunnel Types

Table: Rx Decap Tunnel Types

RX Decap Tunnel Type	Description
RxDecapMPLS3LABEL	Decap MPLS 3 Label, no Control Word present
RxDecapMPLS3LABELCW	Decap MPLS 3 Label, Control Word present
RxDecapMPLS3LABELENTROPYLABEL	Decap MPLS 3 Label + entropy, no Control Word
RxDecapMPLS3LABELENTROPYLABELCW	Decap MPLS 3 Label + entropy, Control Word present

Section 6.13: Precision Time Protocol

Added to bcm_ptp_port_dataset_t

```
typedef struct bcm_ptp_port_dataset_s {
                                              interval */
     uint8 version_number;                 /* Version Number */
     uint8 local_priority;                 /* clock port's local priority */
+    uint8 not_slave;                      /* notSlave config of the clock
port */
 } bcm_ptp_port_dataset_t;
```

Section 6.14: Switch Control

Table: Switch hash fields

Name	Purpose	
bcmSwitchFlexHashFieldCount = 13	Number of fields in this enum.	Modified
bcmSwitchFlexHashField11 = 11,	field 11 in flexible hash RAM table, fed to RTAG7 bin 13, support only UDF type field.	New
bcmSwitchFlexHashField10 = 10,	field 10 in flexible hash RAM table, fed to RTAG7 bin 12, support only UDF type field.	New
bcmSwitchFlexHashField12 = 12,	field 12 in flexible hash RAM table, fed to RTAG7 bin 1, support only UDF type field.	New
bcmSwitchFlexHashField5 = 5,	field 5 in flexible hash RAM table, fed to RTAG7 bin 7, support only UDF type field.	New
bcmSwitchFlexHashField9 = 9,	field 9 in flexible hash RAM table, fed to RTAG7 bin 11, support only UDF type field.	New
bcmSwitchFlexHashField8 = 8,	field 8 in flexible hash RAM table, fed to RTAG7 bin 10, support only UDF type field.	New

bcmSwitchFlexHashField3 = 3,	field 3 in flexible hash RAM table, fed to RTAG7 bin 5, support only UDF type field.	New
bcmSwitchFlexHashField2 = 2,	field 2 in flexible hash RAM table, fed to RTAG7 bin 4, support only UDF type field.	New
bcmSwitchFlexHashField4 = 4,	field 4 in flexible hash RAM table, fed to RTAG7 bin 6, support only UDF type field.	New
bcmSwitchFlexHashField7 = 7,	field 7 in flexible hash RAM table, fed to RTAG7 bin 9, support only UDF type field.	New
bcmSwitchFlexHashField6 = 6,	field 6 in flexible hash RAM table, fed to RTAG7 bin 8, support only UDF type field.	New

Table: Switch Type Values

Name	Description	Value	
bcmSwitchFabricTrunkDynamicPhysicalQueuedBytesMaxThreshold	The maximum threshold, in bytes, used to quantize historical member bytes in physical queue.	0 to maximum fabric trunk port queue size	New
bcmSwitchHashMPLSExcludeReservedLabel	BCM56970 To exclude all reserved labels from hash computation	TRUE/FALSE	New
bcmSwitchFabricTrunkDynamicPhysicalQueuedBytesDecreaseReset	If set, the historical member bytes in physical queue is updated with the instantaneous value if the latter is smaller.	TRUE/FALSE	New
bcmSwitchEcmpDynamicRandomSeed	Seed for RNG used in ECMP Dynamic Load Balancing	0 to 0xFFFFFF	New
bcmSwitchIngressRateLimitMpps	Limit ingress rate in Mega	New	

	Packets Per Second		
bcmSwitchTrunkDynamicRandomSeed	Seed for RNG used in Trunk Dynamic Load Balancing	0 to 0xFFFFFF	New
bcmSwitchFabricTrunkDynamicPhysicalQueuedBytesMinThreshold	The minimum threshold, in bytes, used to quantize historical member bytes in physical queue.	0 to maximum fabric trunk port queue size	New
bcmSwitchEcmpDynamicPhysicalQueuedBytesExponent	The exponent used in the calculation of exponentially weighted moving average of historical member bytes in physical queue.	0 to 15	New
bcmSwitchTrunkDynamicPhysicalQueuedBytesDecreaseReset	If set, the historical member bytes in physical queue is updated with the instantaneous value if the latter is smaller.	TRUE/FALSE	New
bcmSwitchTrunkDynamicPhysicalQueuedBytesExponent	The exponent used in the calculation of exponentially weighted moving average of historical member bytes in physical queue.	0 to 15	New
bcmSwitchMplsNonIpPayloadDefaultEcn	Set the default ECN for L2 MPLS termination flows for non-IP payloads.	0 to 3	New

bcmSwitchTrunkDynamicPhysicalQueuedBytesMinThreshold	The minimum threshold, in bytes, used to quantize historical member bytes in physical queue.	0 to maximum trunk port queue size	New
bcmSwitchEcmpDynamicPhysicalQueuedBytesMinThreshold	The minimum threshold, in bytes, used to quantize historical member bytes in physical queue.	0 to maximum port queue size	New
bcmSwitchHashMultiMoveDepthExactMatch	Maximum levels checked for moving entries in multi bank hash mode for FPEM tables.	>=0	New
bcmSwitchEcnIpPayloadResponsive	If set, for transiting tunnels, use payload IP header's IP protocol field to determine responsive attribute of the packet.	TRUE/FALSE	New
bcmSwitchECMPHashBitCountSelect	Select a 3-bit mask to indicate the most significant bits of the 16-bit ECMP hash value before the modulo function.	0 to 6	New
bcmSwitchLIUdpSrcPortSet	Set the UDP source port of Lawful Interception tunnel	0 to 0xFFFF	New
bcmSwitchFabricTrunkDynamicPhysicalQueuedBytesExponent	The exponent used in the calculation of exponentially	0 to 15	New

	weighted moving average of historical member bytes in physical queue.		
SwitchCrAbort	If working in on demand CR, use this to abort the current transaction.	TRUE/FALSE	New
bcmSwitchFabricTrunkDynamicRandomSeed	Seed for RNG used in HiGig Trunk Dynamic Load Balancing	0 to 0xFFFFFF	New
bcmSwitchTrunkDynamicPhysicalQueuedBytesMaxThreshold	The maximum threshold, in bytes, used to quantize historical member bytes in physical queue.	0 to maximum trunk port queue size	New
bcmSwitchEcmpDynamicPhysicalQueuedBytesMaxThreshold	The maximum threshold, in bytes, used to quantize historical member bytes in physical queue.	0 to maximum port queue size	New
bcmSwitchEcmpDynamicPhysicalQueuedBytesDecreaseReset	If set, the historical member bytes in physical queue is updated with the instantaneous value if the latter is smaller.	TRUE/FALSE	New

```
typedef enum bcm_switch_ecmp_hash_bit_count_e {
    BCM_SWITCH_ECMP_HASH_BIT_COUNT_LOWER_10 = 0, /* Select lower 10 bits of
ECMP hash value to do mudulo operation */`
```

```
    BCM_SWITCH_ECMP_HASH_BIT_COUNT_LOWER_11,
    BCM_SWITCH_ECMP_HASH_BIT_COUNT_LOWER_12,
    BCM_SWITCH_ECMP_HASH_BIT_COUNT_LOWER_13,
    BCM_SWITCH_ECMP_HASH_BIT_COUNT_LOWER_14,
    BCM_SWITCH_ECMP_HASH_BIT_COUNT_LOWER_15,
    BCM_SWITCH_ECMP_HASH_BIT_COUNT_LOWER_16,
} bcm_switch_ecmp_hash_bit_count_t;


typedef enum bcm_switch_object_e {
...
+    bcmSwitchObjectL2EntryMax = 32,     /* Maximum number of L2 entries
+                                           possible. */
...
```

Section 6.15: Trunking (Link Aggregation)

```
typedef struct bcm_trunk_member_s {
...
+    int dynamic_queue_size_weight;      /* Relative weight of queue size
+                                           in determining member quality.
+                                           Specified in percentage. */
} bcm_trunk_member_t;
```

Section 6.16: Tunnel

Table: BCM Layer3 Ingress Interface Flags

Name	Purpose	
BCM_L3_INGRESS_TUNNEL_TERM_ECN_MAP	To set the tunnel termination ecn map id.	New

Table: BCM Tunnel Flags

Name	Purpose	
BCM_TUNNEL_INIT_IPV4_SET_DF	Set DF bit in outer header if inner header is IPv4.	New
BCM_TUNNEL_INIT_SET_DF	Set DF bit in outer header regardless of protocol.	New
BCM_TUNNEL_INIT_GRE_KEY_USE_LB	GRE key presented and include LB key, used with bcmTunnelTypeGRE tunnel types.	New
BCM_TUNNEL_TERM_ETHERNET	Terminate Ethernet header after terminated IP.	New
BCM_TUNNEL_REPLACE	Update existing tunnel.	New
BCM_TUNNEL_TERM_USE_OUTER_PCP	Use outer header PCP.	New
BCM_TUNNEL_INIT_WLAN_MTU	Set MTU for outgoing WLAN tunnel.	New
BCM_TUNNEL_TERM_USE_OUTER_DSCP	Use outer header DSCP.	New
BCM_TUNNEL_TERM_WIDE	Create tunnel terminator eligible for additional data.	New
BCM_TUNNEL_INIT_IP4_ID_SET_RANDOM	Use random starting ID for IPv4 tunnel.	New
BCM_TUNNEL_TERM_USE_TUNNEL_CLASS	Use tunnel class.	New
BCM_TUNNEL_TERM_USE_OUTER_TTL	Use outer header TTL.	New

BCM_TUNNEL_TERM_WLAN_SET_ROAM	Set WLAN ha/fa roaming model.	New
BCM_TUNNEL_TERM_UDP_CHECKSUM_ENABLE	Drop pkts with non-zero UDP checksum.	New
BCM_TUNNEL_INIT_WLAN_TUNNEL_WITH_ID	Create outgoing tunnel with ID.	New
BCM_TUNNEL_TERM_TUNNEL_WITH_ID	Tunnel ID is specified.	New
BCM_TUNNEL_TERM_DSCP_TRUST	Trust incoming tunnel DSCP.	New
BCM_TUNNEL_TERM_WLAN_REMOTE_TERMINATE	Remote termination of WLAN tunnels.	New
BCM_TUNNEL_INIT_WLAN_FRAG_ENABLE	Enable Fragmentation for WLAN Tunnel.	New
BCM_TUNNEL_INIT_IP4_ID_SET_FIXED	Use fixed starting ID for IPv4 tunnel.	New
BCM_TUNNEL_INIT_IPV6_SET_DF	Set DF bit in outer header if inner header is IPv6.	New
BCM_TUNNEL_TERM_TUNNEL_WITH_ID	Tunnel ID is specified.	New
BCM_TUNNEL_INIT_USE_INNER_DF	Copy DF from inner header.	New
BCM_TUNNEL_INIT_WLAN_VLAN_TAGGED	Add the WLAN tunnel VLAN to the outgoing packet.	New
BCM_TUNNEL_TERM_KEEP_INNER_DSCP	Keep inner header DSCP.	New

Section 6.17: VXLAN Management

Table: VXLAN VPN flags

BCM_VXLAN_VPN_TUNNEL_BASED_VNID	Create a VPN supporting network VP per remote VTEP(or tunnel based network VP)	New

```
typedef struct bcm_vxlan_port_s {
     bcm_if_t egress_if;                 /* VXLAN egress object. */
     bcm_switch_network_group_t network_group_id; /* Split Horizon network
group
                                            identifier. */
+    uint32 vnid;                        /* VXLAN packet VNID. */
 } bcm_vxlan_port_t;
```

Section 7: Test Statistics

Section 7.1: How to read the data

The tables below represent a spread of data gathered per device, per suite, and per release. The percentages represent the aggregate rate of failure for that suite when run against all variants of the family of devices.

The below data is not meant to be a precise indication of quality but instead serves as a guideline for improvements release-over-release. Additionally, although some cells show 0% failures, this does not necessarily mean the feature is supported in the device - tests are run to validate the appropriate SDK support even for unsupported features on older devices to ensure graceful handling of all APIs. Finally, some devices have fewer columns listed if they were introduced recently.

Section 7.2: Overview

Each suite listed below is indicative of a specific module. Golden refers to a suite of tests that takes representation across multiple modules and serves as a sanity regression. Each suite contains tests of various types, loosely categorized as follows:

Test Categories	***Description***
Configuration Tests	Tests that verify that each API functions appropriately and can configure the device as expected.
Functionality Tests	Tests that further validate each of the API through functional use often requiring traffic to be run through the system.
Semantic Tests	Tests that ensure that the proper error handling mechanisms are working and users cannot crash the device through the API.

Section 7.2.1: Linux kernel versions used in this release

In SDK 6.5.6, the following Linux kernel versions were used in our development and regression cycles with these main CPUs:

- BCM9XLP208XMC (WRX): 3.10.59
- BCM958625XMC (RSX): 3.6.5

- BCM98548PPCXMC (GTO): 4.4
- BCM98548PPCXMC (XLR): 4.4

Please refer to the Broadcom Network Switching Software Platform Guide for more details about these CPUs.

Section 7.3: Total Tests

The below data represents the number of unique cases for each release.

Note that although a particular test case will execute for each and every chip, it's only counted once.

	sdk-6.5.6	***sdk-6.5.5***	***sdk-6.5.4***
golden	153	153	154
warmboot	3032	3004	2764
auth	17	17	17
bfd	89	86	75
bhh	133	126	61
chip	9	9	9
coe	579	579	591
cosq	768	709	639
custom	7	7	7
ea	108	108	108
eav	19	19	19
extender	49	49	49
fabric	7	7	7
failover	10	10	10
fcoe	37	37	37
field	1470	1435	1415
higigproxy	129	129	129
infra	114	114	114
ipfix	17	17	17
ipmc	114	114	114
l2	337	337	337
l2gre	33	33	33
l3	546	525	522

l3.alpm	550	512	512
link	26	26	26
mim	46	46	46
mirror	173	173	173
misc	21	20	20
mpls	541	483	465
multicast	29	29	28
niv	66	66	65
oam	391	377	358
pkt	44	44	44
port	465	398	374
proxy	38	37	37
ptp	115	115	115
qos	13	13	13
rate	21	21	21
rtag7	43	43	43
rx	25	25	25
ser	157	157	157
stack	117	117	117
stat	403	392	386
stg	42	42	42
switch	197	197	197
time	33	33	33
tlvMsg	13	13	13
trill	47	47	47
trunk	252	232	223
tunnel	133	133	133
subport	31	31	31
vlan	240	240	240
vxlan	219	208	206
wlan	17	17	17
Test Suite Total	12285	11911	11435

Section 7.4: API Test Results

The tables below show percentages of failures for corresponding test suites per SDK release.

Section 7.4.1: All Devices

Note: This section represents aggregate results for all devices in the release.

	sdk-6.5.6	***sdk-6.5.5***	***sdk-6.5.4***
golden	0.04 %	0.1 %	0.1 %
warmboot	0.1 %	0.1 %	0.1 %
bcm.bfd	0.0 %	0.1 %	0.0 %
bcm.bhh	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.coe	0.05 %	0.1 %	0.2 %
bcm.cosq	0.06 %	0.1 %	0.2 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.14 %	0.2 %	0.2 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.01 %	0.1 %	0.1 %
bcm.l2	0.03 %	0.1 %	0.1 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.1 %	0.1 %
bcm.l3.alpm	0.01 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %

bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.02 %	0.1 %	0.1 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.05 %	0.0 %	0.0 %
bcm.multicast	0.1 %	0.1 %	0.1 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.01 %	0.1 %	0.1 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.4 %	0.4 %	0.4 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.1 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.01 %	0.0 %	0.0 %
bcm.stat	0.08 %	0.1 %	0.1 %
bcm.stg	0.02 %	0.0 %	0.0 %
bcm.switch	0.02 %	0.0 %	0.1 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.13 %	0.1 %	0.1 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.16 %	0.0 %	0.0 %
bcm.vlan	0.01 %	0.1 %	0.1 %
bcm.vxlan	0.01 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.2 %	0.2 %	0.2 %

Section 7.4.2: Trident2

	sdk-6.5.6	*sdk-6.5.5*	*sdk-6.5.4*
golden	0.0 %	0.0 %	0.0 %
warmboot	0.09 %	0.1 %	0.1 %
bcm.bfd	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.0 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.01 %	0.1 %	0.1 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.0 %
bcm.l3.alpm	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %

bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.0 %	0.0 %	0.0 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.1 %	0.1 %

Section 7.4.3: Triumph3

	sdk-6.5.6	***sdk-6.5.5***	***sdk-6.5.4***
golden	0.13%	0.1 %	0.1 %

warmboot	0.0 %	0.0 %	0.0 %
bcm.bfd	0.0 %	0.0 %	0.0 %
bcm.bhh	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.0 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.1 %	0.1 %	0.1 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.03 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.03 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.73 %	1.0 %	1.3 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %

bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.04 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.1 %	0.1 %

Section 7.4.4: Katana2

	sdk-6.5.6	***sdk-6.5.5***	***sdk-6.5.4***
golden	0.0 %	0.0 %	0.0 %
warmboot	0.04 %	0.1 %	0.0 %
bcm.bfd	0.0 %	0.0 %	0.0 %
bcm.bhh	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.coe	0.02 %	0.1 %	0.1 %
bcm.cosq	0.0 %	0.0 %	0.1 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %

bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.07 %	0.1 %	0.2 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.08 %	0.1 %	0.1 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.14 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.38 %	0.4 %	0.4 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.1 %	0.1 %	0.1 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %

bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.1 %	0.1 %

Section 7.4.5: Greyhound

	sdk-6.5.6	*sdk-6.5.5*	*sdk-6.5.4*
golden	0.0 %	0.0 %	0.0 %
warmboot	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.coe	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.0 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.01 %	0.0 %	0.0 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.5 %	0.5 %	0.5 %

bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.1 %	0.1 %

Section 7.4.6: Tomahawk

	sdk-6.5.6	***sdk-6.5.5***	***sdk-6.5.4***
golden	0.0 %	0.0 %	0.0 %
warmboot	0.12 %	0.1 %	0.1 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.coe	0.0 %	0.0 %	0.0 %

bcm.cosq	0.02 %	0.2 %	0.3 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.27 %	0.3 %	0.3 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.0 %
bcm.l3.alpm	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.08 %	0.1 %	0.1 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.13 %	0.0 %	0.1 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %

bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.2 %	0.2 %

Section 7.4.7: Trident2+

	sdk-6.5.6	***sdk-6.5.5***	***sdk-6.5.4***
golden	0.0 %	0.0 %	0.0 %
warmboot	0.0 %	0.0 %	0.1 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.0 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.01 %	0.0 %	0.1 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %

bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.01 %	0.0 %	0.0 %
bcm.l3.alpm	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.03 %	0.0 %	0.0 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.12 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %

Test Suite Total	0.1 %	0.1 %	0.1 %

Section 7.4.8: Saber2

	sdk-6.5.6	*sdk-6.5.5*	*sdk-6.5.4*
golden	0.0 %	0.0 %	0.0 %
warmboot	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.coe	0.0 %	0.0 %	0.1 %
bcm.cosq	0.02 %	0.0 %	0.1 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.06 %	0.1 %	0.1 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.18 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %

bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.2 %	0.3 %	0.4 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.1 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.1 %	0.1 %

Section 7.4.9: Hurricane3

	sdk-6.5.6	***sdk-6.5.5***	***sdk-6.5.4***
golden	0.0 %	0.0 %	0.0 %
warmboot	0.08 %	0.1 %	0.1 %
bcm.chip	0.0 %	0.0 %	0.0 %

bcm.cosq	0.0 %	0.0 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.01 %	0.0 %	0.0 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.2 %	0.2 %	0.2 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %

bcm.time	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.1 %	0.1 %

Section 7.4.10: Maverick

	sdk-6.5.6	***sdk-6.5.5***	***sdk-6.5.4***
golden	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.coe	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0%	0.2 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.0 %	0.0 %	0.0 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.1 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %

bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.0 %	0.1 %	0.0 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.1 %	0.1 %

Section 7.4.11: Firebolt5

	sdk-6.5.6	*sdk-6.5.5*	*sdk-6.5.4*
golden	0.0 %	0.1 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.coe	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.2 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.0 %	0.1 %	0.0 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.1 %	0.1 %	0.1 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.13 %	0.1 %	0.3 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %

bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.22 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.2 %	0.3 %	0.2 %

Section 7.4.12: Tomahawk+

	sdk-6.5.6	***sdk-6.5.5***	***sdk-6.5.4***
golden	0.0 %	0.0 %	N/A
bcm.chip	0.0 %	0.0 %	N/A
bcm.coe	0.0 %	0.0 %	N/A
bcm.cosq	0.0 %	0.0 %	N/A
bcm.custom	0.0 %	0.0 %	N/A
bcm.eav	0.0 %	0.0 %	N/A
bcm.extender	0.0 %	0.0 %	N/A
bcm.fabric	0.0 %	0.0 %	N/A
bcm.failover	0.0 %	0.0 %	N/A
bcm.fcoe	0.0 %	0.0 %	N/A
bcm.field	0.13 %	0.1 %	N/A
bcm.higigproxy	0.0 %	0.0 %	N/A

bcm.ipfix	0.0 %	0.0 %	N/A
bcm.ipmc	0.0 %	0.0 %	N/A
bcm.l2	0.0 %	0.0 %	N/A
bcm.l2gre	0.0 %	0.0 %	N/A
bcm.l3	0.0 %	0.0 %	N/A
bcm.link	0.0 %	0.0 %	N/A
bcm.mim	0.0 %	0.0 %	N/A
bcm.mirror	0.0 %	0.0 %	N/A
bcm.misc	0.0 %	0.0 %	N/A
bcm.mpls	0.0 %	0.0 %	N/A
bcm.multicast	0.0 %	0.0 %	N/A
bcm.niv	0.0 %	0.0 %	N/A
bcm.oam	0.0 %	0.0 %	N/A
bcm.pkt	0.0 %	0.0 %	N/A
bcm.port	0.47 %	0.0 %	N/A
bcm.proxy	0.0 %	0.0 %	N/A
bcm.ptp	0.0 %	0.0 %	N/A
bcm.qos	0.0 %	0.0 %	N/A
bcm.rate	0.0 %	0.0 %	N/A
bcm.rtag7	0.0 %	0.0 %	N/A
bcm.rx	0.0 %	0.0 %	N/A
bcm.ser	0.0 %	0.0 %	N/A
bcm.stack	0.0 %	0.0 %	N/A
bcm.stat	0.0 %	0.0 %	N/A
bcm.stg	0.0 %	0.0 %	N/A
bcm.switch	0.0 %	0.0 %	N/A
bcm.time	0.0 %	0.0 %	N/A
bcm.trill	0.0 %	0.0 %	N/A
bcm.trunk	0.0 %	0.0 %	N/A
bcm.tunnel	0.0 %	0.0 %	N/A
bcm.subport	0.0 %	0.0 %	N/A
bcm.vlan	0.0 %	0.0 %	N/A
bcm.vxlan	0.0 %	0.0 %	N/A
bcm.wlan	0.0 %	0.0 %	N/A

Test Suite Total	0.1 %	0.2 %	N/A

Section 7.4.13: Metrolite

	sdk-6.5.6	*sdk-6.5.5*	*sdk-6.5.4*
golden	0.0 %	0.0 %	N/A
bcm.chip	0.0 %	0.0 %	N/A
bcm.coe	0.0 %	0.1 %	N/A
bcm.cosq	0.0 %	0.0 %	N/A
bcm.custom	0.0 %	0.0 %	N/A
bcm.eav	0.0 %	0.0 %	N/A
bcm.extender	0.0 %	0.0 %	N/A
bcm.fabric	0.0 %	0.0 %	N/A
bcm.failover	0.0 %	0.0 %	N/A
bcm.fcoe	0.0 %	0.0 %	N/A
bcm.field	0.0 %	0.0 %	N/A
bcm.higigproxy	0.0 %	0.0 %	N/A
bcm.ipfix	0.0 %	0.0 %	N/A
bcm.ipmc	0.0 %	0.0 %	N/A
bcm.l2	0.0 %	0.0 %	N/A
bcm.l2gre	0.0 %	0.0 %	N/A
bcm.l3	0.0 %	0.0 %	N/A
bcm.link	0.0 %	0.0 %	N/A
bcm.mim	0.0 %	0.0 %	N/A
bcm.mirror	0.0 %	0.0 %	N/A
bcm.misc	0.0 %	0.0 %	N/A
bcm.mpls	0.18 %	0.0 %	N/A
bcm.multicast	0.0 %	0.0 %	N/A
bcm.niv	0.0 %	0.0 %	N/A
bcm.oam	0.0 %	0.0 %	N/A

bcm.pkt	0.0 %	0.0 %	N/A
bcm.port	0.22 %	0.3 %	N/A
bcm.proxy	0.0 %	0.0 %	N/A
bcm.ptp	0.0 %	0.0 %	N/A
bcm.qos	0.0 %	0.0 %	N/A
bcm.rate	0.0 %	0.0 %	N/A
bcm.rtag7	0.0 %	0.0 %	N/A
bcm.rx	0.0 %	0.0 %	N/A
bcm.ser	0.0 %	0.0 %	N/A
bcm.stack	0.0 %	0.0 %	N/A
bcm.stat	0.0 %	0.0 %	N/A
bcm.stg	0.0 %	0.0 %	N/A
bcm.switch	0.0 %	0.0 %	N/A
bcm.time	0.0 %	0.0 %	N/A
bcm.trill	0.0 %	0.0 %	N/A
bcm.trunk	0.0 %	0.0 %	N/A
bcm.tunnel	0.0 %	0.0 %	N/A
bcm.subport	0.0 %	0.0 %	N/A
bcm.vlan	0.0 %	0.0 %	N/A
bcm.vxlan	0.0 %	0.0 %	N/A
bcm.wlan	0.0 %	0.0 %	N/A
Test Suite Total	0.1 %	0.2 %	N/A

Section 7.5: PHY Test Results

Starting with SDK-6.4.8, we have increased our focus and coverage on testing specific PHY and switch combinations in order to improve our quality in this area. The tables below represent specific results from our interoperability and regression testing for the release. We have been continually working to improve our results and coverage in ongoing releases.

Section 7.5.1: SQA External PHY

Switch Device	*External Phy Device*	*Total Tests*	*% Fail*
56639_A0	phy8747_phy8728__10G	198	7.58%
56846_A0	Warpcore_40G/20G/10G	198	1.52%
56850_A2	phy84856_phy84858_10G	198	4.04%
56960_A0	100GFuria	198	1.01%
56960_A0	10GSesto	198	2.02%
56860_A1	phy84856_phy84858_10G	198	1.52%
56860_A0	10GQuadra28	198	0.00%
56860_A0	40GQuadra28	198	0.00%
56860_A0	100GSesto	198	3.54%
56867_A1	phy82764_10G	198	2.02%
56867_A1	phy82764_40G	198	2.02%
56960_B0	phy82864_100G	198	1.52%
56960_B0	phy82864_100G_alt	198	3.03%
56960_B0	phy82864_40GPt	198	1.01%
56960_B0	phy82864_40GPt_alt	198	2.53%
56960_B0	phy82864_10G	198	5.56%
56960_B0	phy82864_10G_alt	198	6.06%
56960_B0	phy82864_11G	198	2.53%
56960_B0	phy82864_11G_alt	198	3.03%
56960_B0	phy82864_42GPt	198	0.51%
56960_B0	phy82864_42GPt_alt	198	2.02%
56960_B0	phy82864_106G	198	1.01%
56960_B0	phy82864_106G_alt	198	2.02%
56960_B0	phy82864_40G2x20	198	4.55%
56960_B0	phy82864_40G2x20_alt	198	5.05%
56960_B0	phy82864_42G2x20	198	2.02%
56960_B0	phy82864_42G2x20_alt	198	2.53%
56960_B0	phy82864_40GMux	198	3.54%
56960_B0	phy82864_50G	198	0.51%
56960_B0	phy82864_25G	198	2.53%
56867_A1	phy82332_100GPt	198	1.52%
56867_A1	phy82332_100G_gbox	198	3.54%
56867_A1	phy82332_106GPt	198	1.52%

56867_A1	phy82332_40G	198	4.04%
56867_A1	phy82332_42G	198	4.04%
56867_A1	phy82332_10G	198	3.54%
56867_A1	phy82332_11G	198	3.54%
56860_A1	phy82332_retimer_1G	198	1.52%
56860_A1	phy82332_retimer_10G	198	4.04%
56860_A1	phy82332_retimer_11G	198	4.04%
56860_A1	phy82332_retimer_40G	198	3.03%
56860_A1	phy82332_retimer_100G	198	1.52%
56765_A0	phy54140_1G_Copper	198	0.51%
56765_A0	phy54140_1G_Fiber	198	0.51%
56565_A0	phy84868_10G	198	0.51%
56160_B0	phy54292_1G	198	1.01%
56960_B0	phy82864_40GMux	147	5.44%

Section 7.5.2: Interop External PHY

Section 7.5.2.1: P2P Suite

Switch Device	*Port Macro*	*Total Tests*	*% Fail*
56560_B0	HG_G40_84328_42G	60	13.33%
56560_B0	XE_G40_84328_10G	148	6.76%
56560_B0	XE_G40_84328_40G	122	4.92%
56560_B0	XE_MT_84757_10G	48	2.08%
56860_A1	XE_QD28_82780_40G	133	3.01%
56860_A1	XE_QD28_82780_10G	167	1.20%
56860_A0	XE_SESTO_82764_40G_MUX	113	12.39%
56860_A0	CE_SESTO_82792_100G_343	222	7.66%
56860_A0	XE_SESTO_82764_40G_PT	184	3.26%
56860_A0	XE_SESTO_82764_10G_PT	168	1.19%
56860_A0	XE_DINO_82332_40G_AN	136	3.68%
56860_A0	CE_DINO_82332_100G_PT_L1_10	204	4.90%
56860_A0	HG_DINO_82332_106G_GB	231	4.76%
56860_A0	CE_DINO_82332_100G_PT	252	0.00%
56860_A0	HG_DINO_82332_11G	86	12.79%
56860_A0	HG_DINO_82332_42G	84	11.90%
56860_A0	CE_DINO_82332_100G_PT_L2_11	204	4.90%
56860_A0	HG_DINO_82332_106G_PT	250	1.60%
56860_A0	XE_DINO_82332_10G	415	0.24%
56860_A0	CE_DINO_82332_100G_GB	338	3.85%
56860_A0	XE_DINO_82332_40G	215	2.79%
56860_A0	CE_DINO_82332_100G_RETIMER	252	0.00%
56860_A0	XE_DINO_82332_10G_RETIMER	360	0.00%
56860_A0	XE_DINO_82332_40G_RETIMER	223	2.24%
56860_A1	XE_G40_84328_10G	148	5.41%
56860_A1	XE_G40_84328_40G	122	4.92%
56860_A0	XE_G28_82322_10G	148	14.19%

56860_A0	XE_G28_82322_40G	122	2.46%
56860_A0	XE_54210	36	2.78%
88675_A0	XE_KOI_8485X	57	1.75%
56960_B0	HG_MADURA_82864_42G_PT	90	0.00%
56960_B0	CE_MADURA_82864_100G_ALT	216	0.00%
56960_B0	XE_MADURA_82864_10G	162	0.00%
56960_B0	HG_MADURA_82864_42G_DUAL_ALT	90	0.00%
56960_B0	XE_MADURA_82864_40G_PT_ALT	162	0.00%
56960_B0	XE_MADURA_82864_25G	48	2.08%
56960_B0	HG_MADURA_82864_42G_PT_ALT	90	0.00%
56960_B0	XE_MADURA_82864_50G	54	0.00%
56960_B0	XE_MADURA_82864_40G_MUX	162	0.00%
56960_B0	HG_MADURA_82864_11G	90	0.00%
56960_B0	HG_MADURA_82864_106G_ALT	120	5.00%
56960_B0	XE_MADURA_82864_50G_ALT	54	0.00%
56960_B0	HG_MADURA_82864_42G_MUX	42	14.29%
56960_B0	XE_MADURA_82864_40G_DUAL	156	7.05%
56960_B0	HG_MADURA_82864_42G_DUAL	66	12.12%
56960_B0	XE_MADURA_82864_25G_ALT	54	0.00%
56960_B0	HG_MADURA_82864_11G_ALT	90	0.00%
56960_B0	XE_MADURA_82864_10G_ALT	162	0.00%
56960_B0	XE_MADURA_82864_40G_DUAL_ALT	162	0.00%
56960_B0	HG_MADURA_82864_106G	80	20.00%
56960_B0	CE_MADURA_82864_100G	178	6.18%
56960_B0	XE_MADURA_82864_40G_PT	162	0.00%
56960_B1	CE_MADURA_82864_100G_RETIMER_ULL	191	5.76%
56960_B1	XE_MADURA_82864_25G_ALT_RETIMER	54	0.00%
56960_B1	XE_MADURA_82864_40G_DUAL_ALT_RETIMER	126	7.94%
56960_B1	XE_MADURA_82864_40G_PT_ALT_RETIMER	162	0.00%
56960_B1	XE_MADURA_82864_10G_RETIMER_ULL	162	0.00%
56960_B1	HG_MADURA_82864_42G_PT_ALT_RETIMER	90	0.00%

56960_B1	XE_MADURA_82864_10G_RETIMER	162	0.00%
56960_B1	HG_MADURA_82864_106G_ALT_RETIMER	48	0.00%
56960_B1	XE_MADURA_82864_50G_RETIMER	54	0.00%
56960_B1	HG_MADURA_82864_11G_ALT_RETIMER	90	0.00%
56960_B1	HG_MADURA_82864_106G_RETIMER	86	16.28%
56960_B1	CE_MADURA_82864_100G_ALT_RETIMER	216	0.00%
56960_B1	HG_MADURA_82864_11G_RETIMER	90	0.00%
56960_B1	XE_MADURA_82864_50G_ALT_RETIMER	54	1.85%
56960_B1	XE_MADURA_82864_25G_RETIMER	54	0.00%
56960_B1	HG_MADURA_82864_42G_PT_RETIMER	90	0.00%
56960_B1	HG_MADURA_82864_42G_DUAL_ALT_RETIMER	90	0.00%
56960_B1	XE_MADURA_82864_40G_MUX_RETIMER	162	0.00%
56960_B1	XE_MADURA_82864_40G_DUAL_RETIMER	138	5.80%
56960_B1	CE_MADURA_82864_100G_RETIMER	200	7.00%
56960_B1	XE_MADURA_82864_10G_ALT_RETIMER	162	0.00%
56960_B1	HG_MADURA_82864_42G_DUAL_RETIMER	66	12.12%
56960_B1	XE_MADURA_82864_40G_PT_RETIMER_ULL	153	0.00%
56960_B1	XE_MADURA_82864_40G_PT_RETIMER	162	0.00%
56960_A0	XE_MADURA_82864_40G_PT	93	0.0%

Section 7.5.2.2: Loopback Suite

Switch Device	*Port Macro*	*Total Tests*	*% Fail*
56560_B0	HG_G40_84328_42G	10	0.00%
56560_B0	XE_G40_84328_10G	10	0.00%
56560_B0	XE_G40_84328_40G	8	0.00%
56560_B0	XE_MT_84757_10G	4	0.00%
56860_A1	XE_QD28_82780_40G	33	0.00%
56860_A1	XE_QD28_82780_10G	55	0.00%

56860_A0	XE_SESTO_82764_40G_MUX	25	80.00%
56860_A0	CE_SESTO_82792_100G_343	47	0.00%
56860_A0	XE_SESTO_82764_40G_PT	30	0.00%
56860_A0	XE_SESTO_82764_10G_PT	30	0.00%
56860_A0	XE_DINO_82332_40G_AN	5	20.00%
56860_A0	CE_DINO_82332_100G_PT_L1_10	56	42.86%
56860_A0	HG_DINO_82332_106G_GB	56	0.00%
56860_A0	CE_DINO_82332_100G_PT	56	0.00%
56860_A0	HG_DINO_82332_11G	25	40.00%
56860_A0	HG_DINO_82332_42G	25	20.00%
56860_A0	CE_DINO_82332_100G_PT_L2_11	56	42.86%
56860_A0	HG_DINO_82332_106G_PT	56	0.00%
56860_A0	XE_DINO_82332_10G	75	22.67%
56860_A0	CE_DINO_82332_100G_GB	63	0.00%
56860_A0	XE_DINO_82332_40G	35	0.00%
56860_A0	CE_DINO_82332_100G_RETIMER	56	0.00%
56860_A0	XE_DINO_82332_10G_RETIMER	40	0.00%
56860_A0	XE_DINO_82332_40G_RETIMER	35	0.00%
56860_A1	XE_G40_84328_10G	40	10.00%
56860_A1	XE_G40_84328_40G	32	6.25%
56860_A0	XE_G28_82322_10G	10	0.00%
56860_A0	XE_G28_82322_40G	8	0.00%
56860_A0	XE_54210	8	0.00%
88675_A0	XE_KOI_8485X	2	0.00%
56960_B0	HG_MADURA_82864_42G_PT	66	9.09%
56960_B0	CE_MADURA_82864_100G_ALT	88	0.00%
56960_B0	XE_MADURA_82864_10G	66	0.00%
56960_B0	HG_MADURA_82864_42G_DUAL_ALT	66	0.00%
56960_B0	XE_MADURA_82864_40G_PT_ALT	66	9.09%
56960_B0	XE_MADURA_82864_25G	22	31.82%
56960_B0	HG_MADURA_82864_42G_PT_ALT	66	9.09%

56960_B0	XE_MADURA_82864_50G	22	18.18%
56960_B0	XE_MADURA_82864_40G_MUX	66	36.36%
56960_B0	HG_MADURA_82864_11G	66	0.00%
56960_B0	HG_MADURA_82864_106G_ALT	88	3.41%
56960_B0	XE_MADURA_82864_50G_ALT	22	18.18%
56960_B0	HG_MADURA_82864_42G_MUX	66	43.94%
56960_B0	XE_MADURA_82864_40G_DUAL	66	24.24%
56960_B0	HG_MADURA_82864_42G_DUAL	66	24.24%
56960_B0	XE_MADURA_82864_25G_ALT	22	9.09%
56960_B0	HG_MADURA_82864_11G_ALT	66	0.00%
56960_B0	XE_MADURA_82864_10G_ALT	66	0.00%
56960_B0	XE_MADURA_82864_40G_DUAL_ALT	66	0.00%
56960_B0	HG_MADURA_82864_106G	88	44.32%
56960_B0	CE_MADURA_82864_100G	88	37.50%
56960_B0	XE_MADURA_82864_40G_PT	66	9.09%
56960_B1	CE_MADURA_82864_100G_RETIMER_ULL	77	46.75%
56960_B1	XE_MADURA_82864_25G_ALT_RETIMER	22	0.00%
56960_B1	XE_MADURA_82864_40G_DUAL_ALT_RETIMER	66	36.36%
56960_B1	XE_MADURA_82864_40G_PT_ALT_RETIMER	66	0.00%
56960_B1	XE_MADURA_82864_10G_RETIMER_ULL	66	27.27%
56960_B1	HG_MADURA_82864_42G_PT_ALT_RETIMER	66	0.00%
56960_B1	XE_MADURA_82864_10G_RETIMER	66	0.00%
56960_B1	XE_MADURA_82864_50G_RETIMER	22	4.55%
56960_B1	HG_MADURA_82864_11G_ALT_RETIMER	66	0.00%
56960_B1	HG_MADURA_82864_106G_RETIMER	88	28.41%
56960_B1	CE_MADURA_82864_100G_ALT_RETIMER	88	2.27%
56960_B1	HG_MADURA_82864_11G_RETIMER	66	0.00%
56960_B1	XE_MADURA_82864_50G_ALT_RETIMER	22	0.00%
56960_B1	XE_MADURA_82864_25G_RETIMER	22	9.09%
56960_B1	HG_MADURA_82864_42G_PT_RETIMER	66	0.00%
56960_B1	HG_MADURA_82864_42G_DUAL_ALT_RETIMER	66	0.00%

56960_B1	XE_MADURA_82864_40G_MUX_RETIMER	66	36.36%
56960_B1	XE_MADURA_82864_40G_DUAL_RETIMER	66	24.24%
56960_B1	CE_MADURA_82864_100G_RETIMER	88	19.32%
56960_B1	XE_MADURA_82864_10G_ALT_RETIMER	66	0.00%
56960_B1	HG_MADURA_82864_42G_DUAL_RETIMER	66	24.24%
56960_B1	XE_MADURA_82864_40G_PT_RETIMER_ULL	55	0.00%
56960_B1	XE_MADURA_82864_40G_PT_RETIMER	66	0.00%
56960_A0	XE_MADURA_82864_40G_PT	33	15.2%

Section 7.6.1: Interop Internal PHY

Section 7.6.1.1: P2P Suite

Switch Device	*Port Macro*	*Total Tests*	*% Fail*
56960_B1	HG_64XD53	108	0.00%
56960_B1	HG_32X106	72	1.39%
56960_B1	HG_128X27	48	6.25%
56960_B1	XE_32X100	136	2.94%
56960_B1	XE_64XD50	204	2.94%
56960_B1	XE_128X25	112	6.25%
56860_A1	HG_32X42	59	3.39%
56860_A1	XE_32X40	87	6.90%
56860_A1	HG_8x100_343	31	3.23%
56860_A1	CE_8X100_343_IEEE	51	17.65%
56860_A1	HG_104x10	21	0.00%
56860_A1	XE_104x10	86	2.33%
56860_A1	CE_8x100_442	51	17.65%
56260_A0	GE_VIPER_12x2P5	36	5.56%

56260_A0	EAGLE_XE_4x10	36	0.00%
56260_A0	GE_VIPER_24x1	39	0.00%
56260_A0	XE_VIPER_6x10	30	0.00%
56260_A0	EAGLE_GE_4x1	12	0.00%
56160_A0	SGMII_TSCE	36	0.00%
56160_A0	QSGMII_TSCE	90	2.22%
56160_A0	QSGMII_QTC	108	0.00%
56160_A0	SGMII_QTC	59	13.56%
56160_A0	SGMII_GPHY	48	0.00%
56160_A0	QSGMII_GPHY	48	0.00%
56960_A0	XE_128X25	89	24.7%

Section 7.6.1.2: Loopback Suite

Switch Device	*Port Macro*	*Total Tests*	*% Fail*
56960_B1	HG_64XD53	33	0.00%
56960_B1	HG_32X106	22	0.00%
56960_B1	HG_128X27	22	0.00%
56960_B1	XE_32X100	22	0.00%
56960_B1	XE_64XD50	33	0.00%
56960_B1	XE_128X25	22	0.00%
56860_A1	HG_32X42	22	27.27%
56860_A1	XE_32X40	22	0.00%
56860_A1	HG_8x100_343	11	0.00%
56860_A1	CE_8X100_343_IEEE	11	0.00%
56860_A1	HG_104x10	11	18.18%
56860_A1	XE_104x10	22	0.00%
56860_A1	CE_8x100_442	11	0.00%
56260_A0	GE_VIPER_12x2P5	28	0.00%

56260_A0	EAGLE_XE_4x10	22	0.00%
56260_A0	GE_VIPER_24x1	30	0.00%
56260_A0	XE_VIPER_6x10	10	0.00%
56260_A0	EAGLE_GE_4x1	10	0.00%
56160_A0	SGMII_TSCE	20	0.00%
56160_A0	QSGMII_TSCE	8	0.00%
56160_A0	QSGMII_QTC	33	0.00%
56160_A0	SGMII_QTC	40	0.00%
56160_A0	SGMII_GPHY	20	5.00%
56160_A0	QSGMII_GPHY	20	0.00%
56960_A0	XE_128X25	22	0.0%

Section 7.7: Static Code Analysis

Starting with SDK 6.5.4, we have upgraded our static analysis code tool to a version with many new checkers and will be working down the backlog of issues. Below shows the current baseline and progress in recent 6.5.x releases:

Section 7.7.1: Unresolved Static Code Analysis Issues

Line of Business	***New baseline as of 3/1/15***	***Open Issues SDK 6.5.6***	***Open Issues SDK 6.5.5***	***Open Issues SDK 6.5.4***	***Open Issues SDK 6.5.3***	***Open Issues SDK 6.5.2***	***Open Issues SDK 6.5.1***	***Open Issues SDK 6.5.0***
DNX	120	63	124	107	104	61	75	125
XGS	90	15	23	37	86	16	27	23
SBX	40	*	*	*	34	1	0	48
SerDes	35	38	25	23	34	12	45	45
Common	45	27	32	42	42	11	16	46
Total	330	143	204	209	300	101	164	301

* As of SDK 6.5.4 SBX quality issues are no longer tracked as part of the release process.

Section 8: Service Impacting Defects

A Service Impacting Defect (SID) is any defect (internal or external) that has high potential to severely disrupt network operations in a deployed system. This section lists the SIDs that have been identified since last release.

Table 34: Resolved Service Impacting Defects

Reference	*Chips*	*Affected Versions*	*Errata Synopsis*	*Details*
No new SIDs identified since last release.				

Section 9: Resolved Issues for 6.5.6

Section 9.1: Resolved Improvements

The table below lists improvement JIRAs that have been added to SDK 6.5.6:

Number	*CSP*	*Chips*	*Release Notes For 6.5.6*
PHY-2528	1067001	56850_A0 56850_A1 56850_A2	Added "NO Firmware download" method
SDK-69905		88470_A0 88670_A0	Added MPLS-TP Section OAM support(Supported the endpoint type bcmOAMEndpointTypeBhhSection) Following are the use cases supported. 1) CCM Processing by OAMP. 2) CCM generation from OAMP. 3) LMM packet stamping by OAM classifier 4) LMM generation from OAMP 5) LMM processing by OAMP and LMR generation. 6) DMM generation from OAMP 7) DMM processing by OAMP and DMR generation. 8) SLM generation by OAMP Following items are not supported in section-OAM: 1) LBM generation 2) LBM processing and LBR generation 3) SLM Processing and SLR generation by OAMP 4) Hierarchical LM (currently only in PRE stage) - only BCM88470-B0
SDK-88994	969667	88470_A0	OAMP PWE injection may be done using BCM_OAM_ENDPOINT_FLAGS2_USE_DOUBLE_OUTLIF_INJ flag.In that case double outlif will be injected and used later in prge. To use this feature: 1. Add soc property: oam_use_double_outlif_injection=1 2. On MEP create use flag2: BCM_OAM_ENDPOINT_FLAGS2_USE_DOUBLE_O

			UTLIF_INJ 3. The interfaces should be given in bcm_oam_endpoint_info_t by intf_id and gport.
SDK-92330	1020605	88670_A0	MPLS: In the previous release, ILM lookup for MPLS LSR service could use the VRF as part of the key of MPLS LSR. In this release, ILM lookup introduces VRF ID as lookup key. 1. add soc property of mpls_contex=vrf; 2. add ILM entry with bcm_mpls_tunnel_switch_create(), set the vpn parameter as VRF id.
SDK-97161	1044042	56440_A0 56440_A1 56440_B0 56450_A0 56450_B0 56450_B1	bcm_multicast_egress_subscriber_delete API would return error when the subscriber queue is already deleted. This has been fixed in this release.
SDK-97893	1037391	88660_A0 88670_A0 88670_B0	SOC property "custom_feature_dedicated_recycle_port" was extended to multiple ports/cores, for example to add port 200 the syntax is: custom_feature_dedicated_recycle_port_200=1
SDK-97902	1045783	88670_A0	BCM_TUNNEL_GRE_KEY_ENABLE flag was added to enable GRE entropy key on normal GRE tunnels.
SDK-98609	1032415	88670_A0 88670_B0	New pre-selector for header format is now supported. bcmFieldHeaderFormatMplsAnyLabelAnyL2L3 supports any number of MPLS labels (1/2/3) for any L2/L3.
SDK-98954	1043135	88670_A0	port_property is now a part of Tunnel Termination "SIP-DIP-VRF" key.
SDK-99755	1054549	AllChips	InterfaceClassPort has been updated to lock SOURCE_TRUNK_MAP_TABLEm during modification, which can prevent intermittent issues during operation.
SDK-100603		88670_A0 88675_A0	TDM Mesh MC mode added to support up to 4 FAP devices per MESH topology.

		This mode supports MESH multicast for TDM traffic only. non TDM traffic can't use MESH multicast replication. New SoC property and API added to support this mode: • In order to activate this mode set "fabric_mesh_multicast_enable=2" • In order to configure multicast replications per ingress port call "bcm_fabric_static_replication_set".
SDK-101240 1060611	88670_A0 88670_B0	BFD: The OAMP TX parameters such as tx_gport, dst_ip_addr, ip_ttl, ip_tos, egress_label.label,egress_label.ttl, egress_label.exp can now be updated using the UPDATE flag during bcm_bfd_endpoint_create.
SDK-101741	88470_A0 88670_A0 88670_B0	Supporting OAM packets with OAM MAC identical to L3 My-MAC.
SDK-101802 1034306	56450_A0 56450_B0 56450_B1	When disabling a subport, the queues associated with the subport are now flushed.
SDK-101844	88470_A0 88670_A0	OAM: New APIs introduced to configure MPLS TP OAM/BFD channel types in both RX and TX direction: bcm_oam_mpls_tp_channel_type_rx_set bcm_oam_mpls_tp_channel_type_rx_get bcm_oam_mpls_tp_channel_type_tx_set bcm_oam_mpls_tp_channel_type_tx_get
SDK-101845	88470_A0	Y1731oPWE with GAL has been supported with new endpoint Type bcmOAMEndpointTypeBHHPweGAL.
SDK-101984 1061982	88675_A0 88675_B0	Which actions/scenarios may trigger the issue: set vid to 0 when calling bcm_port_untagged_vlan_set() What is the impact when the issue occurs: bcm_port_untagged_vlan_set() will return error, because vid 0 is blocked by parameters validation

		What is the impact of the fix: support vid 0 in API bcm_port_untagged_vlan_set(), while vid 0 will still be denied in all other related APIs
SDK-102314 1061824	88270_A0 88470_A0 88670_B0 88680_A0	Added the BCM_MIRROR_DEST_EGRESS_TRAP_WITH_SYSTEM_HEADER flag to the bcm_mirror_destination_{create,get} APIs. The new flag makes packets recycled by egress applications and not forwarded, retain the incoming packet's system headers as received from the fabric, and in case of an Egress MC packet, and stamp the FTMH.OutLIF field with the CUD. The flags can not be used together with the BCM_MIRROR_DEST_EGRESS_ADD_ORIG_SYSTEM_HEADER flag.
SDK-102359	88670_A0 88670_B0 88675_A0 88675_B0	In previous releases, the feature of RX_VGA_RELEASE for bcm_port_phy_control_set was missing implemented. In this release, this feature has been implemented.
SDK-102783 1039239	88670_A0	The command "diag counter" have no statistics of CMIC block's channels TX/RX packets.After this fix, the command contains the all CIMIC channels of TX/RX packets.
SDK-102862 1067562	56850_A0 56850_A1 56850_A2	In previous releases, there was no available API to enable usage of Qgroup limits for a particular queue. In this release, this was fixed by add a new cosq control enum bcmCosqControlEgressUCQueueGroupMinEnable.
SDK-102944 1044825	88670_A0 88670_B0	L2: BCM_FORWARD_ENCAP_ID_EEI_USAGE_ENCAP_POINTER is now supported. This forward encap type should be used when required forwarding result EEI is a pointer to existing egress object. In this case encap_id parameter of bcm_l2_address_add should be the relevant egress object encapsulated with type EEI and usage ENCAP_POINTER.
SDK-103059 1069272	88470_A0	VPLS: In the bcm_mpls_port_get api previously if the failover_port_id is a FEC entry, the validity of FEC pointer will be checked. In this release, this check is removed when customer feature of "conn_to_np_enable" is enabled.

		This change enables user to store any value in learn data of mpls port, even if this value is not a valid FEC pointer.
SDK-103178 1069975	88660_A0	L2: Add BCM_L2_SET_ENCAP_INVALID flag for bcm_l2_addr_t.flags to support adding L2 MAC entry with raw encap ID. When this flag is set, bcm_l2_addr_t.encap_id is added as is in MACT payload, however it is marked as not valid, thus this encap_id will not be used in forwarding, instead, its' value can be pipelined to PMF and used by PMF for various purpose.
SDK-103194 1069873	56960_A0 56960_B0 56960_B1	In previous releases, single packet format can only be associated with single pipe udf entry in pipe local mode. In this release, single packet format is able to be associated with multipipe udf entries in pipe local mode.
SDK-103256	56860_A0	Before this JIRA, a legacy external PHY cannot use the following APIs to configure or retrieve the internal PHY's AN abilities because they were not implemented yet: - phy_portmod_dispatch_advert_ability_set - phy_portmod_dispatch_advert_ability_get - phy_portmod_dispatch_remote_ability_get - phy_portmod_dispatch_local_ability_get Also, phy_portmod_dispatch_an_set, which is used by the legacy external PHY to turn on AN for the internal PHY, needs to be fixed since it always overwrote internal PHY's advert ability with the hard-coded local ability. After the fix of this JIRA, legacy external PHY can set/get internal PHY's AN abilities, and turning AN on for the internal PHY will not change the advert ability to local ability unexpectedly.
SDK-103272	88660_A0 88670_A0 88670_B0	QOS: When doing IP routing, SOC property logical_port_routing_preserve_dscp controls whether to preserve the DSCP field in the forwarding header or not, this is done by dedicated uCode. This change extends the above feature to support IP multicast packets as well.

SDK-103342 1068886	88675_A0 88675_B0	In this release, new counter types has been added for customer to read TFCS,TJBR,TPPKT by bcm_stat_get.
SDK-103596 1071095	56820_A0 56820_B0	Support for SER correction of IPMC_GROUP table is added for BCM56820 device.
SDK-104055 1072156	88670_A0 88670_B0	OAM: BFD endpoint creation improvement - more stability. BFD endpoint creation failure can cause inconsistency and in some cases inability to create more endpoints.
SDK-104188 1074488	56850_A0 56850_A1 56850_A2	In previous releases, scheduling nodes were aligned by 4 at allocation. This has been improved by implementing gport flag BCM_COSQ_GPORT_SCHEDULER_PFC_ALIGN for BCM56850 / BCM56860 series. Scheduler gports created without this flag are not guaranteed to be aligned at 4. To get aligned gport nodes, flag BCM_COSQ_GPORT_SCHEDULER_PFC_ALIGN should be used when creating scheduler gports.
SDK-104581	56960_A0 56960_B0 56960_B1	N/A. This is an improvement. It Add VXLAN feature support for techsupport command for Tomahawk device.
SDK-104696 1076567	56850_A0 56850_A1 56850_A2	In previous releases, API bcm_cosq_stat_get wasn't supported to get Endpoint Queue count. This has been implemented in this release.
SDK-105240	88470_A0	OAM CFM: In the previous release, this type "OAM-o-Eth-o-PWE-o-MPLS-o-ETH" packet will trap to CPU to process as OAM packet, but it can't be forwarded as data packet. In this release, this issue has been addressed, this type of packet will forward as data packet by default,and user can also use SOC "custom_feature_oam_downmep_pwe_classification" to trap it as OAM packet.
SDK-105460 1079831	88470_A0	For label extended encapsulation feature, packet goes out will take additional label shim with it. An problem occurs when packet output through COE port that COE shim doesn't exist. A new program is added to support it and enabled by soc property "custom_feature_mpls_extend_over_coe".

SDK-105714 1079865	56565_A0 56565_B0	In the earlier version, pivot route was not being set with default route setting when bucket was split. The issue has been resolved with the correct setting.
SDK-105963 1080836	56860_A0 56860_A1	In previous releases, SDK did not support mapping a range of Q-in-Q VLAN (Outer VLAN range & Inner VLAN range) to a specific SVP for the VxLAN access port. This has been addressed in this release.

Section 9.2: Resolved Issues

The table below lists all defects resolved in SDK 6.5.6:

Number	*CSP*	*Chips*	*Release Notes For 6.5.6*
PHY-2455	1050570	56860_A0 56860_A1	HG40 support has been added to Sesto (82792).
PHY-2561	1061119	AllChips	Preemphasis set API has been updated for BCM82864.
PHY-2613	1060568	56960_A0	In the previous release, Link scan can't read the actual link status of external PHY Furia(82381) during the warm boot. The issue has been fixed in this release.
SDK-88792	1000311	56860_A0	Document the up-to-date polarity configuration properties in SDK, including the newly implemented physical port based one.
SDK-89393	1006217	88660_A0	In "diag field res", the prefix structure was not initialized. Because of that we were getting in many iterations of junk output. Now, initialization is added, and the problem no longer exists.
SDK-94853	1026567	88950_a0	In previous releases, the fabric port may return incorrect medium type through API . In this release, this issue has been fixed.
SDK-95881	1030707	88650_A0 88670_A0	TX: when sending a packet in 'TX', the CRC cannot be found at the FCS field in 'RX',it's next to the FCS field instead. Fixing it by putting the calculated CRC to correct field.
SDK-97807	1046827	56860_A0 56860_A1	In previous releases, when the RIOT was enabled, the API bcm_l2_station_add invoked without BCM_L2_STATION_UNDERLAY might throw BCM_E_RESOURCE even if there were resources, which was incorrect. In this release, this issue has been addressed.
SDK-98791	1051173	56860_A0 56860_A1	Missing check for all supported 100G interfaces, and as the result there are times we did not reset TSCE before changing modes. We now check for speed instead to ensure TSCE got reset for all applicable interfaces.
SDK-99066		56260_B0	Viper does not have Ucode and no DSC -(Digital Conditioning Block) as a result dsc commands are not supported

SDK-99605	1021123	AllChips	Introduced 4 new config property to use physical port based lane swap and polarity and they are phy_chain_tx_polarity_physical phy_chain_rx_polarity_physical phy_chain_tx_lane_map_physical phy_chain_rx_lane_map_physical
SDK-99643	1038042	88650_A0 88650_B0 88650_B1 88660_A0 88670_A0 88670_B0	In field processor, when removing an entry from direct extraction DB, the FEM instruction is not cleaned properly. in addition, other programs can be affected by this operation. This is Fixed.
SDK-100666		56160_B0	Modified HR3 B0 TS and BS PLL setting for 50MHz internal CLK
SDK-100676	1049383	88670_A0	Meter issues: bcm_policer_color_decision_se\get not configured\read the correct entry. Fixed.
SDK-101294	1060563	56460_A0 56460_B0	Enabling the Transmit Clock Alignment (TCA) function to phase align all four lane's clocks so that frontend data path logic can be properly sampled into the configured PMA lane.
SDK-101325	1061448	88470_A0	meter logic read meter buckets status every clock cycle and there is no access to CPU to IMP/IEP blocks. Solution: Add bubble for QAX device, blocks IMP/IEP. (will be bubbled only when CPU ask access)
SDK-101446	1059324	56334_A0 56334_B0	Fixed the unstable issue for L3 host conflict get API that was due to non-initialization of variable used in internal function _bcm_xgs3_l3_bucket_get().
SDK-101596		88670_B0	In BCM88{3,6}{7,8}x support was added to control whether forwarded copies of packets are dropped in ingress when the snoop/mirror copy is dropped due to FIFO overflow. The new API may not be called under traffic: int bcm_switch_control_set(unit, bcmSwitchMirrorSnoopForwardOriginalWhenDropped, 0/1);
SDK-101978	1062757	56450_A0 56450_B0	In previous releases, if customer used non-volatile RAM and enabled auto-sync for warmboot, the API

		56450_B1	such as bcm_port_enable_set would take more time. In this release, this issue has been fixed.
SDK-102078		88375 88470 88650 88660 88670 88675	In previous version Interdigitated connectivity between VOQ and flow isn't set correctly, therefor not all flows in flow group will send credits. This issue is fixed.
SDK-102121	1062281	AllChips	In previous releases, VP-LAG gport couldn't be processed in bcm_port_tpid_set()/get()/add(). This has been fixed in this release.
SDK-102447	1062316	88660_A0 88670_A0	API can convert any ingress SW trap_id to HW trap_id, the API: bcm_rx_trap_sw_to_hw_id_map_get(unit, flags, sw_trap_id, *hw_trap_id) int unit (IN) - device number uint32 flags (IN) - reserved int sw_trap_id (IN) - Software trap id int *hw_trap_id (OUT) - Hardware trap id which is sent in FHEI header For virtual traps must be called after: 1) bcm_rx_trap_type_create() 2) bcm_rx_trap_type_set() 3) bcm_l2_cache_set()
SDK-102485	1066659	88470_A0	PMF actions bcmFieldActionClassDestSet and bcmFieldActionClassSourceSet did not work. This is fixed
SDK-102521	1065439	56440_A0 56440_A1 56440_B0	Intr_enable should be reset after processing the interrupt so that it can wait for the next interrupt.
SDK-102587	1066867	88670_A0 88670_B0	When ILKN was set to non-default speed and user was trying to add the other ILKN port with a shared quad - the new port was changing the speed of the shared quad back to the default speed so the link of the existing ILKN port was falling down. The fixed code doesn't change the speed of a shared quad during port_attach.
SDK-102775	1057845	88660_A0	OAM: Prevented resource leak when replacing a server UpMEP with ccm_period = 0

SDK-102793	1066930	88670_A0	In previous version EGQ to CPU traffic getting stuck after soft-reset (when soft reset is done during traffic). Fixed
SDK-102805		88670_A0	In egress PMF in Jericho, the counting is done via ACE-table by bcmFieldActionStat. The redirect action also was changing ACE-type, thus conflicting when used together with counting. Now the action bcmFieldActionRedirect does not change ACE-type and so it can be used together with counting.
SDK-102809	1065248	88670_A0	ACL: When user set PMF action "bcmFieldActionStat" for redirection and counter with SOC property "custom_feature_lsr_cnt_egress=1", this action didn't take effect. The issue affected redirection and counting base on PMF. The bcmFieldActionStat action can be set to HW after the fix through modifying the SOC checking condition.
SDK-102812		88670_A0 88675_A0	bcm_fabric_bandwidth_profile_set was limited to 288 profiles instead of 296. Limitation was fixed.
SDK-102857		56860_A0 56860_A1	In previous releases, when creating the multicast group for VLAN service with subtending ports, BCM_MULTICAST_TYPE_SUBPORT type multicast group couldn’t work. In this release, it has been resolved by attaching BC_IDX,UUC_IDX and UMC_IDX to this type multicast group.
SDK-102949	1068303	56640_B0	In previous releases, when port type was COMBO_MODE, FEC was only configured for one lane in WarpCoreC. In this release, this issue has been fixed.
SDK-102982	1068644	56160_A0 56160_B0	XG_PLL1 configuration is modified for External Phy.
SDK-103018	1055834	88670_B0	In previous releases the Endian checking of KBP was incorrect, which may cause KBP cannot boot up correctly when using a little Endian CPU. The issue has been fixed in this release.
SDK-103025	1063737	88670_A0	when custom feature vt_tst2 is set, bcmFieldQualifyOuterVlanId is mapped to initial_vid qualifier, for all stages at init. since there is no support for VT/TT stages in qualifiers mapping in SDK, ingress-PMF mapping is also performed, hence qualifying upon inital VID inseted og outer VID.

SDK-103044	1058022	88670_B0	OAM: standard MEP configuration can affect also default MEP configuration. This means that under certain circumstances traps created by the endpoint will remain.
SDK-103174	1067633	88650_A0 88660_A0	VLAN: ITAG_TPID will be set 0x88e7 when MIM is disabled. A problem occurs when send 1tag packet which ether-type is 0x88e7, it will be recognized 2tags This change is to don't set ITAG_TPID when MIM is disabled, the tag-format will be correct and traffic can be passed.
SDK-103279	1059243	88375_B0	The API bcm_mirror_destination_get does not consider all parameters. Due to this, the users cannot get all flag and encap value through this API. Now flag and encap value can be got through this API.
SDK-103362	1070074	53416_A0	This release will fix the bcm_port_ability_advert_get() retruning 1G, even 1G is not advertised in CL73 AN mode.
SDK-103388	1070039	56960_A0 56960_B0 56960_B1	In previous releases, there is a race condition that global resource of memscan being accessed by multi threads during SDK initialization, resulting in device crash. Now in this release, the issue is fixed.
SDK-103398	1070878	56960_A0 56960_B0 56960_B1	In previous releases, same MTP for ingress and egress mirror was not allowed when port was in SAF mode on Tomahawk B1. In this release, same MTP for ingress and egress mirror is allowed when port is in SAF mode on Tomahawk B1
SDK-103469	1065439	56450_A0 56450_B0 56450_B1 56840_A0	Intr_enable should be reset after processing the interrupt so that it can wait for the next interrupt.
SDK-103480	966123	88660_A0	Out-AC: When PON DSCP remark feature was enabled, a data entry will be allocated to keep the new DSCP value using the VLAN port creation APIs. A problem occurs when after the data entry is destroyed and trying to create a new VLAN port with

			the same ID. This is due to that internal resources was not released properly during VLAN port destroy. This change fixes above issue, VLAN port with the same ID can be created once it is destroyed.
SDK-103521	1069280	88375_B0 88670_B0	SER: XOR memory protection was not implemented. A problem occurs when MACT is accessed by two different APPs at the same time. The error cannot be recovered when XOR bank or other SP banks happened SER error. Then the interrupt can not be cleared. The issues describe above affected XOR memory SER protection. XOR memory SER protection can work correctly after the fix.
SDK-103536	1066916	88675_B0	In previous releases, the SDK disabled and enabled Mac frequently when resetting BCM8867X, which may cause some corrupt packets in buffer when traffic was on-going. In this release, the corrupt packets can be drained out when the above scenario happens.
SDK-103537	1068240	88660_A0	OAM: Prevented resource leaks when deleting endpoints created on Ingress only LIFs
SDK-103553	1067991	56640_A0 56640_A1 56640_B0	bcm_tr_mpls_update_vp_nh is used to update dependent VP Nexthops based on the data from the current HW egress next_hop entry. When the entry type is 1, MAC DA was be replaced in HW in bcm_tr_mpls_egress_entry_mac_replace. Since bcm_tr_mpls_egress_entry_mac_replace is called after bcm_tr_mpls_update_vp_nh, dependent VP NextHops were not getting updated with proper MAC DA. Changed code to call bcm_tr_mpls_update_vp_nh only after HW entry is updated with required changes.
SDK-103653	1072152	56960_A0 56960_B0 56960_B1	Lower bound value of range checker set to 0 whenever range checker is destroyed.

SDK-103663	1072175	56640_A0 56640_A1 56640_B0	When bcm_l3_egress_create is called with replace and MPLS label as invalid. SDk will delete any existing data related to current vc swap index in the HW entry. But for LSP entry, VC swap index will be pointing to 0 (which is a reserved an entry). SDK was trying to retrieve/delete SW info related to vc swap index entry. Since it was reserved entry, related info was not present in SW database. So API was returning failure. Added check to skip updating SW database when vc swap index is found to be reserved.
SDK-103686	1072751	56340_A0	Writing to register was wrongly using the cos value when it should have used the calculated startq value.
SDK-103692	1070597	56460_A0 56460_B0	The hardcoded TOD port map values have been removed to allow for backward compatibility.
SDK-103694	1068911	56860_A0 56860_A1	In previous releases, multicast packets might be dropped in Cut-Through mode because MMU_ENQ_LOGICAL_PORT_TO_PORT_AGG_MAP and MMU_ENQ_REPL_PORT_AGG_MAP were not set. In this release, this has be resolved by programming them correctly.
SDK-103715	1067645	56344_A0	notSlave variable added in portDS in compliance with G8275.1
SDK-103731	1051670	88650_A0 88660_A0 88670_A0 88670_B0 88675_A0 88675_B0	RX:In some specific OAM scenarios, non-trapped packets received by CPU were not parsed correctly due to some uninitialized fields of structure soc_pkt_t. In this release, it has been fixed.
SDK-103748	1070958	56850_A0 56850_A1 56850_A2	In previous releases, bcm_switch_control_port_set didn't support bcmSwitchColorSelect for proxy ports. In this release, the issue has been fixed.
SDK-103774	1071394	88375_A0	MPLS PORT: An error has been set, forbidding the setting of mpls_port.egress_label in ingress only mode with the flag BCM_MPLS_PORT2_LEARN_ENCAP. This is because in this case, we learn the encap id, so the VC label should be empty.

SDK-103786	1072276	56151_A0	Removed non-existent XGXS_PLL register configuration from HR2 reset to avoid the core and added support for xgxs_lcpll_xtal_refclk soc property
SDK-103837	1052125	56860_A0 56860_A1	In this release, the SDK now keeps track of usage information of the external phy. When all the lanes of the external phy are detached from the PM, it will need to reinitialize and re-download. Customer will need to detach all the lanes associated with that external phy in order for that phy to be re-probed and downloaded.
SDK-103862		88675_A0	SER: The print of special handing SER was not right. The print of special handing SER may be mislead the users. These handing actually do nothing, but it prints software reset. The issue describes above affected the understanding of SER action. The print is right after the fix.
SDK-103908		88675_A0	Dynamic Port: When removing ETH port and adding ILKN, the port might get stuck. This is due to some ETH specific configuration which was not cleared before removing the port. The issue is fixed.
SDK-103951	1072708	88470_A0	NIF: QSGMII ports failed to be enabled/disabled by BCM shell command. This has been fixed.
SDK-103953	1073667	88470_A0	COE: Packet can be forced from one port to another port after setting bcm_port_force_forward_set(). A problem occurs when port vlan COE feature is enable. Packet injected from CPU port didn't have correct system header. New program selection is added to handle this case.
SDK-103968	1073484	88375_A0 88375_B0	In previous version when adding new channel to active ILKN interface other ports on the same interface were affected (momentary egress queues build up). This issue was fixed.
SDK-104017		88675_A0	Soc property "fc_inband_intlkn_calender_length" should be added with "fc_inband_intlkn_mode" and "fc_inband_intlkn_calender_llfc_mode" for the case

			that Jericho connect with KBP and use the NBI to process the flow control from KBP.
SDK-104054	1074443	56640_B0	There are three things fixed in this JIRA 1) disable per lane clock for all 8/10 bit aggregated speed mode 2) allow tx remote fault for all the aggregated HG mode. 3) RX fifo reset before CDR locks fix for all 8/10 bit aggregated mode
SDK-104062	1073019	56540_A0 56540_B0	When the port is down the RRPKT are updated as part of the ibodsync thread update but recovery sequence will not kick in for a disabled port. On updating the counters updated, when the port comes up Ibod thread will see no difference and the recovery doesn't run and the packets get corrupted. rrpkt counter are not updated when recovery sequence is not run.
SDK-104067	1066156	88670_A0 88670_B0	In previous releases, some error logs may be printed during initialization when only one KBP was connected to BCM88670. This has been fixed in this release.
SDK-104088	1074322	88470_A0	In previous version when using bcmCosqControlBandwidthBurstMax control to configure burst during traffic, once burst was configured to 0 (disable traffic) it couldn't be recovered (also when re-set to higher value). This issue is fixed.
SDK-104103	1071075	88470_A0	In previous version there was a problem with bcm_cosq_gport_sched_set with COMPOSITE_SF2 when first call uses cosq different then 0 and than configuring cosq 0. Issue is fixed.
SDK-104116	1074319	56846_A0	SDK didn't update S/W cache for the physical instances THDO_OFFSET_0A/B when writing data to the THDO_OFFSET_0 table. In this release, the issue has been fixed.
SDK-104156		88375_A0 88375_B0	bcm_cosq_src_vsqs_gport_add() crashes when using ILKN over fabric. Fixed.
SDK-104161		56042 56040 56041	In previous releases, the PLL divider was changed to the wrong value when 1G speed was set. This issue has been fixed in the release.

		56340M 56342M 56344M 5634X 56548 56547	
SDK-104185	1074756	56760_A0 56760_B0	This release addresses issues related to calculations for temperature readings of PVTMON
SDK-104205	1074430	56860_A0 56860_A1	Register file didn't have affected memory marked as SER_ECC_CORRECTABLE and the SWFLAGS were not set to enable them to be cachable (SOC_MEM_FLAG_CACHABLE). Received new register file from hardware, compiled and verified SER recovered memory properly.
SDK-104221	1050219	88650_A0 88660_A0	There was inconsistency in message string passing arad_interrupt_handle_<par/1ecc/2ecc>() functionality between Arad+ in Jericho; the 4th parameter in arad_interrupt_handles_corrective_action() is "msg" while it should be "special_msg" following the same logic as Jericho In this release, all the arad_interrupt_handles_corrective_action() calls in the code have been changed to local variable "special_msg" instead of "msg".
SDK-104261	1074297	AllChips	Possible leak can happen on multi chip board. Fixed it with array of pointers.
SDK-104294		88675_A0 88675_B0	Outband FC : Configuring LLFC for OOB for ILKN port. API won't populate correctly the Interface index in RX OOB table. The issue is FC reception over OOB calendar won't trigger correct ilkn interface LLFC. The Interface index in RX OOB table can be added correctly after the fix.
SDK-104302	1074527	56850_A0 56850_A1 56850_A2	In previous releases, if flex Base Index reference count for Accounting Object was none-zero, API bcm_stat_group_destroy() would return BCM_E_FAIL. In this release, this return value has been corrected to BCM_E_BUSY.

SDK-104350	1075408	56850_A0 56850_A1 56850_A2	The issue occurred due to access of NULL soc_control structure pointer. Fix implemented to check soc pointer for NULL, before using the MACRO which in turn access the structure members. If soc pointer is NULL interrupt handler will return immediately.
SDK-104351	1075198	88660_A0	OAM: Prevent freeing the OEM-1 table when oam_classifier_advanced_mode=2, if the OEM-1 entry is being shared by multiple MEPs
SDK-104476	1073510	56260_A0 56260_B0 56460_A0 56460_B0	Issue : User was creating MIP's with non-zero ccm period and was unable to delete the LMEP entry created with MIP flag post warm boot. Fix: Oam endpoint validation check has been updated, to prevent creation of MIP's with non-zero ccm period.
SDK-104497	1075764	88660_A0 88670_A0	when adding entries to TCAM, error code returned when TCAM out of resources was -1 (general). Error code changed in this case to out of resource to indicate that the TCAM is full
SDK-104528	1065607	AllChips	In this release, we updated sal_sem_take API to validate timeout parameter and return non-zero value when negative timeout value is specified.
SDK-104571	1062185	56960_A0 56960_B0 56960_B1	In previous releases, byte counter was treated as a 32 bits counter and the stat in higher bits was ignored. In this release, the issue has been fixed by determining correct counter length as 34 bits and taking all counter stat into calculation.
SDK-104583	1074356	56440_B0 56450_A0 56450_B0 56450_B1	Added a check to not allow configurations which will have burst value more than 16.449Kbits vlaue. Burst value should be less than or equal to 16.449 Kbits.
SDK-104658	1076395	88670_B0	For Jericho B0 and above, tunnel initiator create would return an error "out of memory" when replacing a tunnel. With this fix, out of memory won't occur when replacing the tunnel.
SDK-104674	1076495	56260_A0 56260_B0	Problem: During MPLS label replace operation, when the new and the old label are same, then the mpls entry was getting deleted.

			Solution: Deletion of mpls entry is skipped when the replaced label is same as original label.
SDK-104699		88375_A0 88670_A0	NIF: Fixed incorrect QSGMII ports initialization sequence which may lead to transmitting CRC error packets occasionally.
SDK-104714	1055248	88660_A0	In the previous release, in api of bcm_l3_egress_create(), FEC resource will be allocated before checking the parameter validation. In this release, it will check the validation of destination parameter first. New FEC resource could be allocated only when the destination parameter is valid.
SDK-104718		5626X 5644X 5645X 5646X 5645X	For a multicast group having internal and external extended queue members, information about internal and external copies was getting corrupted after addition/deletion of members. This has been corrected.
SDK-104826	1074991	88375_B0	In previous releases, TPID could not be returned by API bcm_tunnel_initiator_get. This issue has been fixed.
SDK-104828	1074874	56460_A0 56460_B0	For SDK re-boots (without hard reset), and re-loads the FW, SDK re-programs the Master and Serdes LCPLL FBDIV control values to the known(i.e, power on reset) values in the chip reset API.
SDK-104910		88675_A0	removing of trunks from egress MC group is now possible
SDK-104935		88670_A0	The PRGE program selection of ARAD_EGR_PROG_EDITOR_PROG_MPLS_EXTENDED_ELI_JER and ARAD_EGR_PROG_EDITOR_PROG_MPLS_EXTENDED_JER were updated to include roo presence.
SDK-104941	1064327	88660_A0	Relevant for ARAD/+ only. In L3, when uRPF was enabled, packets were dropped even if the forwarding lookup in LPM DB got a hit with a valid forwarding decision. This happened due to incorrect setting of the relevant FLP program when INCLUDE_KBP was not used in compilation. This erroneous configuration is now fixed.

SDK-104944	1076225	56860_A0 56860_A1	In previous releases, VFI memory lock was used to protect VXLAN VPN creation, which was abused and unnecessary. In this release, this lock has been removed to prevent deadlock.
SDK-104991	1067557	56565_A0 56565_B0	This release addresses issues related to HSP tree creation on Firebolt5
SDK-104992	1041443	56860_A0	Fixed port driver implementation for i2c read/write operations
SDK-105016	1075678	56743_A0 56744_A0 56750_A0 56750_A1 56750_A2	In previous releases, bcmSwitchIngParseL3L4IPv4 and bcmSwitchIngParseL3L4IPv6 switch control options were not supported. This has been addressed in this release.
SDK-105056	1074596	56860_A0 56860_A1	In the previous releases, bcm_vlan_translate_action_add can not be used when the match key is trunk and vlan. In this release, the issue doesn`t exist.
SDK-105072	1077711	56850_A0 56850_A1 56850_A2	In previous releases, the max mtu size in KNET was rx buffer size plus RCPU header size. This was incorrect, Because this may result in packet data used RCPU header space. In this release, the issue has been fixed
SDK-105173	1078260	56850_A0 56860_A0 56960_A0 56970_A0	In previous releases, it would not return BCM_E_EXISTS when the API bcm_l2_addr_add was invoked to add an entry which was same as the entry added by the API bcm_l2_cache_set with BCM_L2_CACHE_L3. This has been fixed in this release.
SDK-105179	1076367	88670_B0	ILKN BurstShort now follows the same logic for ELK port as for non-ELK ports. Use SOC property ILKN_BURST_SHORT to set the BurstShort parameter (per device). BurstShort should be a multiplier of 32 and lesser or equal to BurstMax/2. Default value for BurstShort is 32B.
SDK-105180	1063219	56546_A0 56546_B0	Runt Counters statys unmodified in hardware for ibod to capture corruption everytime irrespective of stat clear command issued

SDK-105183	1077135	88670_A0 88670_B0 88675_A0 88675_B0	"diag cosq voq detailed" doesn't work for multicast queues. Fixed. The API bcm_cosq_gport_traverse() traverse over all queues. Up until now it started with queue 4. From now on it will start from queue 0, which is multicast queue in the default application. Pay attention that if you used this function and assumed all queues are unicast queues, you might get errors.
SDK-105186		88470_A0	DNX parser correctly takes into account system headers (UDH)
SDK-105239	1078496	56850_A0 56850_A1 56850_A2	In previous releases, PFC cannot be passed to the system. This has been addressed by enabling feature pgw_mac_pfc_frame for BCM56850.
SDK-105249	1078526	56850_A0 56850_A1 56850_A2	In previous releases, when called command "l3 defip show", the class id value wasn't displayed correctly if it was more than 63. This has been fixed in this release.
SDK-105250	1077473	88680_A0	IP source binding: IP source binding enables binding a source IP address to a service flow (a VLAN port or a FEC pointer in the reverse path), it requires customer to create a PMF group to compare SIP lookup result from forwarding lookup stage to InLIF ID or InLIF learn data (FEC pointer), and drops the packet if they are not matched. A problem occurs when PON DSCP remarking feature is enabled, in which case a new data entry is created, SDK will fail to resolve the correct InLIF ID or FEC pointer from PON LIF gport when installing a L3 source bind entry, thus in the PMF stage it will never return a match and drop the packet. This change fixes the above issue, so that SDK will resolve to the correct InLIF ID or FEC pointer. PMF compare group will return a match when the source IP is bound.
SDK-105269	1074656	88670_A0 88670_B0	MPLS: Calling bcm_stat_lif_counting_profile_set when soc property default_logical_interface_mpls_termination_explicit_null is set resulted in error.

SDK-105270	1075448	88675_A0 88675_B0	In previous release network headers were not updated for EVPN non ROO packets. In this release this issue is resolved. In addition, FRR ttl was updated to be calculated similarly as in previous version and is set to 255.
SDK-105272	1078577	56860_A0 56860_A1	In previous releases, the size of src_trunk packed within ct_rx_tunnel_pkt_pack() was 1-byte, it should be 2-byte to support for up to 1k trunk groups. This has been fixed in this release.
SDK-105284	1078107	88670_A0 88670_B0	EGQ threshold adjustment was done for ILKN less then 150Gbps. This adjustment should slightly improve EGQ prioritization in oversubscribe scenarios.
SDK-105292	1079078	56743_A0 56744_A0 56750_A0 56750_A1 56750_A2	In previous releases, bcmSwitchHashMimUseTunnelHeader0 and bcmSwitchHashMimUseTunnelHeader1 switch control options were not supported. This has been addressed in this release.
SDK-105306	1060568	56960_A0	In previous releases, Link scan can't read the actual link status from external PHY Sesto(82764) during the warm boot. The issue has been fixed in this release.
SDK-105313	1074297	AllChips	Possible leak can happen on multi chip board. Fixed it with array of pointers.
SDK-105372		88675_B0	SER: The XOR process of the table PPDB_A_FEC_SUPER_ENTRY_BANK was not right. A problem occurs when the table PPDB_A_FEC_SUPER_ENTRY_BANK happened SER error.. The error cannot be recovered when XOR bank or other SP banks happened SER error. The issues describe above affected the table PPDB_A_FEC_SUPER_ENTRY_BANK memory SER protection. The table PPDB_A_FEC_SUPER_ENTRY_BANK memory SER protection can work correctly after the fix.
SDK-105393	1075558	56860_A0	AN slice, tech ability, FEC ability, pause ability were programmed for the non AN interfaces, which is fixed now.

SDK-105426		5654X 5664X	On TR3 runt threshold setting corrected for HG ports after init
SDK-105434	1067947	88675_B0	Which action cause this issue: Customer redirect packet to KNET Netif and check packets format with tools like dumptcp, dumptcp shows detailed content of packets, and found that the packet format is not correct. What does issue look like: 1) when the packet was not vlan-tagged, "0806 0001", the arp protocol field, was stripped as vlan tag. 2) in x86 devices, ftmh header was parsed incorrectly, followed 10bytes PPH header and FHEI extension header were sent to KNET netif and then tcpdump. with the unexpected 10 bytes dune header, the packets seems in wrong format. 3) KNET CRC strip error. the last 4 bytes of packets were always stripped by KNET. on XGS devices, packets are sent to KNET with CRC, while on Dune devices, CRC has been stripped already. What impaction of this issue. And how it fixed in this release: all these three issues cause packets in wrong format. 1) check to make sure that when packets were vlan-tagged, KNET strips vlan tag. 2)ftmh parsing error in x86 devices due to byte order. litter-endian and big-endian are not well considered, all are handled as big-endian, while Customer's x86 is litter-endian. improve 32bytes-swap operation. 3)on dune devices, do not strip CRC
SDK-105446	1078834	53457_A0	In previous releases, the qsgmiie remote loopback get function was missing; In this release, this issue has been fixed.
SDK-105448	1079774	56860_A0 56860_A1	Enable encap mode change when the port has AN enabled.

SDK-105503	1078755	56860_A0 56860_A1	Fixed the incorrect data type used to store the Phy core number in warmboot state.
SDK-105516	1075877	88670_A0 88670_B0	QOS-remark: Added better allocation-manager for OutLIF profiles to enable bcmPortPreserveDscpEgress feature with other OutLIF profile applications.
SDK-105520		88470_A0	In previous version WRED min threshold is always 0 (can't be configured). This issue is fixed.
SDK-105561	1079264	56640_A0 56640_A1 56640_B0	For HSP Node, Hw index, populated during gport_add, was getting errored during gport_attach. Same has been fixed.
SDK-105607	1080266	56640_A0 56640_A1 56640_B0	When Min Limits changed, the shared limits were recalculated for all SPIDs. This was the root cause of the issue. This was fixed by changing shared limits only for the relevant SPID.
SDK-105610	1080025	56340_A0	Corrected multicast queue offset derivation in bcm_cosq_control_set/get API for Triumph3/Helix4.
SDK-105616	1078187	56340_A0	Corrected the size calculation when copying data from L2_ENTRY table to shadow table in l2_sync thread.
SDK-105697	1080876	88660_A0	COE port can be enabled and disabled through bcm_port_control_set(). A problem occurs when set operation for CPU port. Assertion will appear if disable for bcmPortControlExtenderEnable type. Counting of port is removed to resolve this issue.
SDK-105710		88470_A0 88680_A0	OAM: OAM packet could be trapped to the CPU because of an outer LIF MEP. Even though there is no MEP on the actual LIF. for example, PWE OAM packets might be trapped for PWE o LSP o ETH packet while an endpoint is only set on the LSP.
SDK-105744	1074406	56340_A0	Issue seen due to multithreading race condition. Locks were added to avoid race condition.
SDK-105763	1074641	88670_B0 88675_B0	For Ingress PMF, the qualifier bcmFieldQualifyInnerEtherType was taking the 16 bits

			of Ethertype directly after the SA of the inner Ethernet, assuming a non-tagged header. Now tagged inner Ethernet headers are supported, by taking the Ethertype bits relative to the next header start with a negative offset.
SDK-105823	1081373	88470_A0	In QAX, configuring ILKN interface with TDM channels may fail(return an error) due to wrong register access. Fixed in this release.
SDK-105883	1080811	88675_B0	bcm_tunnel_initiator_create() didn't deallocate the created global and local lifs in case the tunnel initiation failed due to sip/ttl/tos allocation failure.
SDK-105892	1079344	88470_A0 88680_A0	The memory table FDT_IPT_MESH_MC was not cleared properly at init time in QAX and JER+, it can cause ECC errors when sending mesh traffic. In this release, this problem has been fixed.
SDK-105915	1080576	88675_A0 88675_B0 AllChips	creating a route on a interface with a LAG being a part of a VLAN with LAG ID which works got all lag group IDs.
SDK-105930	1081505	56860_A0 56860_A1	Fix memory leakage issue for TD2P to improve the memory performance during flex.
SDK-105936	1081437	56960_A0 56960_B0 56960_B1	EFP counter also need to be read in sync mode
SDK-105938	1074982	56860_A0 56860_A1	In previous releases, for BCM5686x devices, whenever a tpid other than the default tpid was created, reference count of default tpid was decreased once but was not increased during deletion. During repeated creation and deletion, this reference count became negative resulting in error. In this release, this has been fixed by increasing default tpid reference count upon deletion of tpid thus providing support for repeated creation and deletion of tpid on a port.
SDK-105996	1081669	88650_B1 88660_A0 88670_B0	In bcm_petra_field_stat_create the stat_id parameter was not passed down properly. Now it is fixed, with removing the dereferencing and changing the variable to be a pointer.
SDK-106012	1081727	AllChips	Fixed register read with address (instead of symbol) in BCM shell.
SDK-106050	1066399	88660_A0	When enabling OAM statistics(OAM_STATISTICS_PER_MEP_ENABLED) on Arad+ in OAM Server-client model, on the client device it is required to define recycle port for all

			UP-MEPs: custom_feature_oam_up_mep_client_recycle_port=[port]
SDK-106110	1076087	56460_A0 56460_B0	Fixed the loss count profile and opcode control profile creation using the endpoint_action set APIs
SDK-106118		88650_A0 88660_A0 88675_A0	in previous version, When you change a CL type under-traffic(remove the CL, change the type and reattach it), this CL get stuck, fixed.
SDK-106218	1081357	88670_A0 88670_B0	When advanced preselector is created in non-ingress stage, bcm_field_presel_create_stage_id API should be used, otherwise the stage will not be correctly assigned.
SDK-106229	1051449	56450_A0	The virtual lock that was acquired was not being relinquished in a couple of return paths, this is now fixed.
SDK-106258		88950_a0	Modified the SKU feature restrictions to support value "REPEATER" for SOC property fabric_device_mode for FE3200 SKUs.
SDK-106259	1078163	88375_A0 88375_B0	ILKN1/3/5 did not work when set to low priority (using bcm_port_nif_priority_set API). This issue is fixed. NOTE: ports sharing a serdes quad should have the same priority.
SDK-106260		88670_A0 88670_B0	SER: Interrupt mask show value was not right. The mask value of the command "listmem ser-action" or "show interrupts" is opposite. A problem occurs when the table PPDB_A_FEC_SUPER_ENTRY_BANK happened SER error.. The error cannot be recovered when XOR bank or other SP banks happened SER error. The issues describe above affected the users understanding. The show can work correctly after the fix.
SDK-106329	1083102	56850_A0 56850_A1 56850_A2	Ensure that the Linux BDE DMA memory pool is allocated within the 32-bit addressing space across all processor architectures.

SDK-106380	1082934	56820_A0 56820_B0	For BCM56820 device, index for IPMC_GROUPx used in SER correction is calculated from error pointers available of both the slices.
SDK-106383	1084287	88470_A0	In the previous release, native eve AC could not be deleted using the api of bcm_petra_port_match_delete(); from this release, this issue is fixed.
SDK-106387	1082965	56260_A0 56260_B0	Prevented leakage in EGR_MPLS_VC_AND_SWAP_LABEL_TABLE by deleting old swap entry if valid during mpls add call
SDK-106407	1082722	88670_B0 88675_B0 88680_A0	ClamShell is unsupported in GDDR5. When SDK try to enable Clamshell for GDDR5, it will be rejected, and exit initialization.
SDK-106415	1081980	88670_B0 88675_B0	L3: in bcm_l3_egress_create EGRESS_ONLY replace, failover_id field is now ignored as failover_is is not an egress property in this api.
SDK-106418	1077538	88650_B1 88660_A0 88670_A0	in previous version, calling some PP APIs that use pp_port_info_set could cause CGM port profile modification , fixed.
SDK-106494	1083052	56340_A0 56340M_A0 AllChips	Corrected multicast queue offset derivation in bcm_cosq_control_set/get API for Triumph3/Helix4 for cosq controls bcmCosqControlEgressMCQueueMinLimitBytes and bcmCosqControlEgressMCQueueSharedLimitBytes
SDK-106544	1078040	56640_A0 56640_A1 56640_B0	Warm boot recover H/w index to logical index mapping for RMEP is missing Solution: Add the H/w index to logical index mapping for RMEP
SDK-106617	1085038	56860_A0 56860_A1	Problem: Missing check for int_pri against allowed priorities in bcmi_xgs5_bfd_endpoint_create Solution: Add Validation for int_pri value passed to configurating cos queue for the tx port
SDK-106618	1085369	88660_A0	In the LEM/EM SER handling, wrong index of tables was used when reading LEM/EM memories, because index was valid only in the first array, other array

			index was not taken into consideration. In this release, this problem has been fixed.
SDK-106744	1084142	88660_A0	In BCM_INIT_ST, a race condition was experienced. If we have thread, which is already waiting on our lock instead of the winner, it should receive an error during its next time slice at the point of the wait, and that will propagate back to the caller as BCM_E_INTERNAL. To avoid the race condition, the competitive processes are now synchronized.
SDK-106778	1085723	88470_A0	In the previous release, MPLS RAW function will remove extra bytes when UDH is enabled. In this release, this has be fixed.
SDK-106782	1086102	56560_A0 56560_B0 56850_A0 56850_A1 56850_A2 56860_A0 56860_A1	The offset for qualifier bcmFieldQualifyDstGPort is set correctly during initialization of ingress qualifiers.
SDK-106784	1086156	53400_A0	Fix the port block valid check issue to bring up the TSC port in bcm53400 and bcm88675 combined platform.
SDK-106791	1084690	56760_A0 56760_B0	This release addresses issues related to initialisation of init_speed used during port probe
SDK-106839	1085283	88470_A0	In previous releases the default Rx Max frame size of Unimac was 1518B. This has been changed to 16360B in this release.
SDK-106860	1086350	88470_A0	In this release, for mpls tunnel whose action is bcm_mpls_egress_action_nop, it could be removed successfully. using the api of bcm_mpls_tunnel_initiator_clear().
SDK-106873	1086633	88660_A0 88670_A0 88670_B0	OAM: OAMP interrupt handler can lose events and leak memory in transient situations.
SDK-106922	1085239	56240_B0	For some EFP groups, selcodes are not recovered when efp group doesn't have any entries. Recovery code is enhanced to handle no-entries scenerio.

SDK-106930	1086918	88670_B0	SER: The corrected XOR value may be not right.. A problem occurs when the length of XOR memory is less than 20 bits. The corrected value is wrong. The issues describe above affected XOR memory SER protection. XOR memory value can work correctly after the fix
SDK-106948	1086821	AllChips	Only one buffer was allocated for punted packets. Hence when the punted packets were added to dcb chains under scaled scenarios, if the dcb had not yet been dequeued by DMA engine, there was a possibility of next punt packet corrupting the same buffer (which is put into DCB) with its information. This was the reason for duplicated packets and therefore missing packets. We have fixed this issue by using a pool of buffers for punt packets. The number of buffers in the pool can be set by new custom soc property as given below: custom_feature_num_punt_buffers=32 Please set the above value to 32. Any increase in that number will affect scaling in future.
SDK-106980	1086799	56850_A0 56850_A1 56850_A2	For each entry, atmost four mirrorIngress/MirrorEgress can be configured. But warmboot sync and recovery code handles only first instance of the action. While trying to recover other instance of the same action, data is incorrectly updated with the first instance. Enhanced sync and recovery to consider all the four instance of MirrorIngress/MirrorEgress Action.
SDK-107025	1087076	56840_A0	Fixed by compressing all metering pipes during field group compress.
SDK-107041		88680_A0	ILKN ports using shared first quad but defined on different cores will cause init \ dynamic port failure. Issue is fixed.
SDK-107076	1085745	88375_A0 88670_A0	In previous release, API bcm_vlan_port_find might return error if it worked with egress failover. It has been fixed.

SDK-107113		88680_A0	In previous releases BCM88680 cannot connect to BCM52311 with 6 lanes. This has been supported in this release.
SDK-107114	1087344	56640_A0 56640_A1 56640_B0	Corrected the procedure for checking and marking whether Fail-over index is in use, during warmboot,
SDK-107123	1086443	88670_A0 88670_B0	In L3/IPMC, when adding an entry to the LPM (KAPS), using the BCM_L3_FLAGS2_RAW_ENTRY or BCM_IPMC_RAW_ENTRY respectively, some payload values were rejected due to incorrect verification. This is fixed now as all 20 bit payload values are allowed in raw payload mode.
SDK-107162	1087809	56960_A0 56960_B0 56960_B1	In previous releases, bcm_port_vlan_member_set and bcm_vlan_gport_add didn't support mpls port on TH. In this release, bcm_port_vlan_member_set and bcm_vlan_gport_add do support mpls port on TH.
SDK-107177	1083052	56340_A0 56340M_A0 AllChips	Corrected multicast queue offset derivation in bcm_cosq_control_set/get API for Triumph3/Helix4 for cosq controls bcmCosqControlEgressMCSharedDynamicEnable
SDK-107329	1084600	88670_B0	Once a cold boot is invoked and the detach function is called, the sw db which represents the SCH_SWITCH_CIR_EIR_IN_DUAL_SHAPERS will be free so on next boot these array will be initialized again and the SCH_SWITCH_CIR_EIR_IN_DUAL_SHAPERS will be writen
SDK-107450	1089087	56640_A0 56640_A1 56640_B0	Key type value17 is updated in MPLS_ENTRY_EXTDm memory, to get mpls_index and mpls_entry_extd_entry. This fix is applicable only for TR3 devices.
SDK-107476	1088305	56460_A0 56460_B0	LM/DM packet priority will be taken from Loss/Delay_add packet priority.
SDK-107489	1083020	88670_B0	bcm_oam_init() was failing at memory allocation when BCM hw reload done multiple times. Memory leakage was found and fixed.

Section 10: Unresolved Issues for 6.5.6

The following open Urgent priority issues remain unresolved in version 6.5.6 of the SDK.

Number	*CSP*	*Chips*	*Errata for 6.5.6*
PHY-2019	976013	56340_A0	SDK build fails when compiled with bmacsec for iproc devices
SDK-86322		56340_A0	10G SFP port flapped on a Helix 4 based customer platform (with ext PHY) when the peer interface status was brought from "down" to "up"
SDK-91219	1014281	88670_A0 88670_B0 88950_A0	Using bcm_port_phy_control get for Falcon RX settings leaves uc stopped
SDK-103675	1072355	56854_A0 56854_A2	SDK6.5.3 and SDK6.5.4, Some of 20G ports couldn't tx and rx with loopback=mac on the specific config
SDK-103800		88670_A0	ILKN SerDes can't be locked if enable CL72 at 6.4.11 while it's OK at 6.4.10
SDK-103803	1066867	88670_A0 88670_B0	When ETH is set to non-default speed and user is trying to add the other ETH port sharing the same quad - the new port is changing the speed of the shared quad back to the default speed so the link of the existing port is falling down.
SDK-104806	1075344	56860_A0 56860_A1	Need BCM API support for inner vlan range (INNER_VLAN_RANGE_IDX) in double tagged interface

Section 11: Device and Platform Support

This section has been removed from the release notes. For the full list of Broadcom switch and PHY devices supported in the SDK, please reference the file SDK-6.5.6-Device-Matrix.xlsx in the RELDOCS directory in the release package.

Section 12: Compatibility

Section 12.1: Broadcom Task Engines (BTE) Firmware Compatibility Matrix

The following table shows compatibility between different versions of SDK and Firmware releases. Please refer to the appropriate Network Switching SDK Firmware release notes publication (56XX0_88XX0_FW-RNxxx-R) for the indicated version below.

	SDK-6.5.6	SDK-6.5.5	SDK-6.5.4	SDK-6.5.3	SDK-6.4.11	SDK-6.4.10
4.3.3	BCM56270 BCM56760 BCM56565					
4.3.2		BCM56270 BCM56965 BCM88470				
4.3.1			BCM56760 BCM56560 BCM56565			
4.3.0				BCM56760 BCM56560 BCM56565		
4.2.7					BCM56230 BCM56260 BCM56460 BCM56445 BCM56450	BCM56260 BCM56460 BCM56445 BCM56450

Section 12.2: BMACSEC SDK Compatibility Matrix

Switch SDK Release	*BMACSEC Release*
6.5.0	4.14
6.5.1	4.14
6.5.2	4.15
6.5.3	4.15
6.5.4	4.15
6.5.5	4.15
6.5.6	4.16

Section 12.3: PHY Firmware Compatibility Matrix

The following table identifies changes in PHY firmware for newer PHY devices.

PHY Core	**6.5.2 Firmware Versions**	**6.5.3 Firmware Versions**	**6.5.4 Firmware Versions**	**6.5.5 Firmware Versions**	**6.5.6 Firmware Versions**
BCM84861	00.00.10	00.00.10	01.00.00	01.00.00	01.00.00
BCM84864	00.00.10	00.00.10	01.00.00	01.00.00	01.00.00
BCM84868	00.00.10	00.00.10	01.00.00	01.00.00	01.00.00
BCM84858	01.02.10	01.03.02	01.03.02	01.03.02	01.03.02
BCM84856	01.02.10	01.02.10	01.03.02	01.03.02	01.03.02
BCM84744	0x0132	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132	0x0130

	(B0/C0)			(B0/C0)	(B0/C0)
BCM84757	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)
BCM84328	R027	R029	R029	R029	R029
BCM82322/28 B1	0xF	0xF	0xF	0xF	0xF
BCM82780	0x23	0x23	0x23	0x23	0x23
BCM82752	0x23	0x23	0x23	0x23	0x23
BCM82758	0x23	0x23	0x23	0x23	0x23
BCM82764	D011	D013	D017	D017	D017
BCM82790	D011	D013	D017	D017	D017
BCM82792	D011	D013	D017	D017	D017
BCM82796	D011	D013	D017	D017	D017
BCM82381	D011	D013	D013	D013	D013
BCM82209	D011	D013	D013	D013	D013
BCM82073	D011	D013	D013	D013	D013
BCM82864	N/A	N/A	D00C	D00C	D00C
BCM82332	N/A	N/A	D006	D008	D008
Eagle	D10F_05	D10F_OD	D10F_0D	D10F_0D	D10F_13
Falcon	D10A_07	D10B_07	D10B_0C	D10B_0E	D10B_13

Section 13: SDK Externally Licensed Software Components

The SDK contains a number of third-party externally licensed software components. This appendix contains information regarding these components, the license for each of these components, and where these components are used in SDK.

Component	*Origin*	*Location in Source Tree*
EDITLINE	/afs/athena.mit.edu/contrib/sipb/src/editline	src/sal/appl/editline
LIBXML2	http://xmlsoft.org/downloads.html	src/shared/libxml
ED Editor	USENET comp.sources.misc Volume 9, Issue 36	src/appl/diag/edline.c
CINT	http://www.gnu.org/software/bison/	src/appl/cint/cint_parser.[ch]
BIGDIGITS	David Ireland, copyright (c) 2001-11 by D.I. Management Services Pty Limited <www.di-mgt.com.au>	src/soc/dpp/SAND/Utils/sand_u64.c
APIMODE	http://www.gnu.org/software/bison/	src/appl/diag/api/api_grammar.tab.[ch]
VxWorks	Wind River Systems, Inc.	systems/vxworks

Section 13.1: EDITLINE License terms and conditions

```
This package was obtained from the following location, and was modified
for purposes of inclusion into the SOC diagnostics shell.

Removed files:
        MANIFEST Make.os9 Makefile os9.h sysos9.c testit.c unix.h

Added files:
        sysvxworks.c Makefile

Changed functionality:
        Merged unix.h into editline.h
        M-P and M-N now behave like tcsh.
        list_history(count) routine displays history
```

```
        Commented out completion
        Changed rl_complete and rl_list_possib into caller-settable
                global functions
        Don't ring bell on TAB if word is already complete

--------------------------------------------------------------------------------

Index of /afs/athena.mit.edu/contrib/sipb/src/editline

[     ] Name                    Last modified     Size  Description
  ------------------------------------------------------------------------

 [DIR]  Parent Directory       11-May-99 03:40      -
 [   ]  MANIFEST               07-Jul-97 11:20     1k
 [   ]  Make.os9               07-Jul-97 11:20     1k
 [   ]  Makefile               01-Sep-97 00:34     2k
 [   ]  complete.c             07-Jul-97 11:20     4k
 [   ]  editline.3             07-Jul-97 11:20     5k
 [   ]  editline.c             07-Jul-97 11:20    25k
 [   ]  editline.h             07-Jul-97 11:20     2k
 [   ]  os9.h                  07-Jul-97 11:20     1k
 [   ]  sysos9.c               07-Jul-97 11:20     1k
 [   ]  sysunix.c              07-Jul-97 11:20     3k
 [   ]  testit.c               07-Jul-97 11:20     1k
 [   ]  unix.h                 07-Jul-97 11:20     1k

  ------------------------------------------------------------------------

$Revision: 1.7 $

This is a line-editing library.  It can be linked into almost any
program to provide command-line editing and recall.

It is call-compatible with the FSF readline library, but it is a
fraction of the size (and offers fewer features).  It does not use
standard I/O.  It is distributed under a "C News-like" copyright.

Configuration is done in the Makefile.  Type "make testit" to get
a small slow shell for testing.

This contains some changes since the posting to comp.sources.misc:
    -   Bugfix for completion on absolute pathnames.
    -   Better handling of M-n versus showing raw 8bit chars.
    -   Better signal handling.
    -   Now supports termios/termio/sgttyb ioctl's.
    -   Add M-m command to toggle how 8bit data is displayed.
The following changes, made since the last public release, come from
J.G. Vons <vons@cesar.crbca1.sinet.slb.com>:
    -   History-searching no longer redraws the line wrong
    -   Added ESC-ESC as synonym for ESC-?
    -   SIGQUIT (normally ^\) now sends a signal, not indicating EOF.
    -   Fixed some typo's and unclear wording in the manpage.
    -   Fixed completion when all entries shared a common prefix.
    -   Fixed some meta-char line-redrawing bugs.
```

```
Enjoy,
        Rich $alz
        <rsalz@osf.org>

 Copyright 1992,1993 Simmule Turner and Rich Salz.  All rights reserved.

 This software is not subject to any license of the American Telephone
 and Telegraph Company or of the Regents of the University of California.

 Permission is granted to anyone to use this software for any purpose on
 any computer system, and to alter it and redistribute it freely, subject
 to the following restrictions:
 1. The authors are not responsible for the consequences of use of this
    software, no matter how awful, even if they arise from flaws in it.
 2. The origin of this software must not be misrepresented, either by
    explicit claim or by omission.  Since few users ever read sources,
    credits must appear in the documentation.
 3. Altered versions must be plainly marked as such, and must not be
    misrepresented as being the original software.  Since few users
    ever read sources, credits must appear in the documentation.
 4. This notice may not be removed or altered.
```

Section 13.2: LIBXML2 - XML C parser terms and conditions

Package was obtained from http://xmlsoft.org/ and is used by diagnostics tool for miscellaneous input/output tasks

This README is part of SDK under src/shared/libxml and is as follows:

```
/*
 * $Id$
 *
 * $Copyright: (c) 2011 Broadcom Corporation
 * All Rights Reserved.$
 */

This package was obtained from http://xmlsoft.org/downloads.html
(ftp://xmlsoft.org/libxml2/libxml2-2.7.2.tar.gz)
and was modified for purposes of inclusion into the SOC diagnostics shell.

Only certain portion of package was included in SDK in 2 places:
    Under srs/shared/libxml
        chvalid.c, config.h, dict.c, encoding.c, entities.c, error.c
        globals.c, hash.c, libxml.h, list.c, Makefile, parser.c
        parserInternals.c, SAX2.c, threads.c, tree.c, uri.c, valid.c
        xmlIO.c, xmlmemory.c, xmlsave.c, xmlstring.c, xmlunicode.c
    Under include/shared/libxml
        catalog.h, chvalid.h, debugXML.h, dict.h, DOCBparser.h
        encoding.h, entities.h, globals.h, hash.h, HTMLparser.h
        HTMLtree.h, list.h, parser.h, parserInternals.h, pattern.h
        relaxng, SAX2.h, threads.h, tree.h, uri.h, valid.h, xinclude.h
        xlink.h, xmlautomata.h, xmlerror.h, xmlexports.h, xmlIO.h
```

```
       xmlmemory.h, xmlmodule.h, xmlregexp.h, xmlsave.h, xmlstring.h
       xmlunicode.h, xmlversion.h, xpath.h, xpathInternals.h, xpointer.h

No functionality was changed, but there were modifications to match SDK
requirements

Copyright

Except where otherwise noted in the source code (e.g. the files hash.c,
list.c and the trio files, which are covered by a similar licence but
with different Copyright notices) all the files are:

 Copyright (C) 1998-2003 Daniel Veillard.  All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is fur-
nished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FIT-
NESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL THE
DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CON-
```

```
NECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.


Except as contained in this notice, the name of Daniel Veillard shall not
be used in advertising or otherwise to promote the sale, use or other deal-
ings in this Software without prior written authorization from him.
```

Section 13.3: ED Editor License terms and conditions

```
ed - standard editor
  ^^
    Authors: Brian Beattie, Kees Bot, and others

 Copyright 1987 Brian Beattie Rights Reserved.
 Permission to copy or distribute granted under the following conditions:
 1). No charge may be made other than reasonable charges for reproduction.
 2). This notice must remain intact.
 3). No further restrictions may be added.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  TurboC mods and cleanup 8/17/88 RAMontante.
  Further information (posting headers, etc.) at end of file.
 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
 Modification log:
 25Aug92  (W.Metzenthen) Changed malloc() call to calloc() in makebitmap()
          to remove bugs under Linux. Changed a few '^' to the correct '~'.
          General tidying. Recognize Linux via the __linux__ symbol.
          Main change based upon suggestion by Wolfgang Thiel.
 07Sep99  Changed large amounts of stuff to simplify  --Curt McDowell
```

Section 13.4: CINT parser license terms and conditions

The C code for the CINT parser was generated by using GNU Bison parser generator from the file cint_grammar.y CINT is an optional diagnostic tool that can be included in your system by adding CINT to the FEATURE_LIST in SDK compilation flags.

Removed files:

None

Added files:

None

Changed functionality:

None

```
------------------------------------------------------------------------------
/* A Bison parser, made by GNU Bison 2.4.1.  */

/* Skeleton implementation for Bison's Yacc-like parsers in C

      Copyright (C) 1984, 1989, 1990, 2000, 2001, 2002, 2003, 2004, 2005, 2006
   Free Software Foundation, Inc.

   This program is free software: you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation, either version 3 of the License, or
   (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program.  If not, see <http://www.gnu.org/licenses/>.  */

/* As a special exception, you may create a larger work that contains
   part or all of the Bison parser skeleton and distribute that work
   under terms of your choice, so long as that work isn't itself a
   parser generator using the skeleton or a modified version thereof
   as a parser skeleton.  Alternatively, if you modify or redistribute
   the parser skeleton itself, you may (at your option) remove this
   special exception, which will cause the skeleton and the resulting
   Bison output files to be licensed under the GNU General Public
   License without this special exception.

   This special exception was added by the Free Software Foundation in
   version 2.2 of Bison.  */
```

```
/* C LALR(1) parser skeleton written by Richard Stallman, by
   simplifying the original so-called "semantic" parser.  */
```

Section 13.5: BIGDIGITS license terms and conditions

```
Contains BIGDIGITS multiple-precision arithmetic code originally
written by David Ireland, copyright (c) 2001-11 by D.I. Management
Services Pty Limited <www.di-mgt.com.au>, and is used with
permission.

David Ireland and DI Management Services Pty Limited make no
representations concerning either the merchantability of this software
or the suitability of this software for any particular purpose. It is
provided "as is" without express or implied warranty of any kind. Our
liability will be limited exclusively to the refund of the money you
paid us for the software, namely nothing. By using the software you
expressly agree to such a waiver. If you do not agree to the terms, do
not use the software.
```

Section 13.6: APIMODE parser license terms and conditions

The C code for the APIMODE parser was generated by using GNU Bison parser generator from the file api_grammar.y APIMODE is an optional diagnostics shell interface that can be included in your system by adding APIMODE to the FEATURE_LIST in SDK compilation flags.

See “CINT parser license terms and conditions” for the Bison licence.

Section 13.7: Wind River Systems license terms and conditions

See WRS_LICENSE.pdf contained in each systems/vxworks subdirectory.