Release Notes
For
Switch Software Development Kit

SDK 6.5.7

© 2016 Broadcom. All rights reserved

Broadcom®, the pulse logo, Connecting everything®, the Connecting everything logo, Avago Technologies, BroadR-Reach®, BroadSync®, Flexport, Hex-PHY™, HiGig™, HiGig2™, HiGig+™, StrataXGS®, Tomahawk™, Warpcore™, XGS™, and XGS Core®, are among the trademarks of Broadcom and/or its affiliates in the United States, certain other countries and/or the EU. Any other trademarks or trade names mentioned are the property of their respective owners. Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

Table of Contents

Section 1: About This Document

These are the Release Notes for the Broadcom Network Switching Software Development Kit Release 6.5.7.

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-PG657-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***
BCM88470	BCM88470 B0	32x12.5GE + 16x25GE Ethernet Switch
BCM56965	BCM56965 A1	32x 100GbE/64x 40GbE/128x 10GbE Multilayer Switch

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
BCM53570	BCM53570 A0	High Port Count Integrated Switch With 100-FX/1G/2.5G/5G/10G/25G-Capable SerDes Lanes
BCM56170	BCM56170 A0	High Port Count Integrated Switch With 100-FX/1G/2.5G/5G/10G/25G-Capable SerDes Lanes
BCM56965	BCM56967 A1	32x 100GbE/64x 40GbE/128x 10GbE Multilayer Switch
BCM88270	BCM88270 A0	28x2.5GE + 8x10.3GE Ethernet Switch

Section 3.3: Supported PHYs

Family	*Devices*	*Description*	*Switch qualified against*
BCM84848	BCM84888 A0	10G/5G/2.5G/1000/ 100BASE-T Transceiver	BCM56860

Section 3.4: Preview PHYs

Family	*Devices*	*Description*	*Switch qualified against*
BCM54190	BCM54190 A0/B0	Octal SGMII Copper Gigabit Ethernet Transceiver	

Section 4: New Features per Device

Section 4.1: BCM56970 (Tomahawk2) Beta support

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

Section 4.1.1: Features supported

The table below shows the status of supported features in SDK 6.5.7. These features have been tested on actual silicon in this release unless otherwise noted. Most features are considered Beta in this release, meaning testing has progressed beyond the bringup stage. Overall DVAPI test results are around 99.8% passing but some individual modules may be higher.

Table 1: SDK Feature Status

Feature	***Status***	***Note***
L2	Beta	
L3	Beta	
VLAN	Beta	
Port	Beta	Support 16nm Serdes
Flex Port	Beta	Legacy lane set API deprecated
STG	Beta	
Trunk	Beta	
Mirror	Beta	
Switch Controls	Beta	
Legacy MPLS	Beta	
VxLAN	Beta	
Mac in Mac	Beta	
Proxy	Beta	
Higig Proxy	Beta	
Legacy Stats	Beta	
Flex Counters	Beta	ALPM flex counter support added
Qos Assignment	Beta	
Rate	Beta	

Meters	Beta	
Policer	Beta	
IPMC	Beta	
Multicast	Beta	
Legacy Subports	Beta	
VP-VLAN Membership checks	Beta	
NIV	Beta	
Port Extender	Beta	
Trill	Beta	
NAT	Beta	
VPLAG	Beta	
Legacy Protection Switching	Beta	
Field Processor	Beta	New actions and qualifiers added, see API guide for more details
UDF	Beta	
System and Network level congestion	Beta	
Packet Tx/Rx	Beta	
Base MMU	Beta	
COSQ	Beta	Added buffer ID support for MMU threshold and WRED settings
BST	Beta	Added buffer ID support; added 2 new BST objects
RIOT	Preview	
SER	Beta	
Visibility: Aggregation Group Monitor (AGM)	Beta	
Visibility: Packet Trace	Beta	
LPM	Beta	
ALPM	Beta	Enhanced ALPM utilization
RTAG7 flex hash	Beta	Support all 13 RTAG7 BINs to extract from UDF
Resilient Hash	Beta	
ECMP and Hierarchical ECMP	Beta	
UFT	Beta	Optimized breadth-first hash reordering algorithm
DLB	Preview	
DGM	Preview	
TCB	Beta	

Packetized stats	Beta
Packet timestamping	Beta
MPLS enhancements	Beta

Section 4.1.2: New and enhanced features supported on BCM56970

The following features are new to BCM56970 and the table below describes the new support in this release.

Table 2: New features supported

Feature	***Note***
DLB over ECMP, and 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
MPLS Enhancements	● MPLS ECN Marking ● MPLS Label Swap ● MPLS Segment Routing ● MPLS FRR (Fast Reroute) Lookup ● MPLS 5 Labels Parsing

Section 4.2: BCM56965 (Tomahawk+) family updates

Section 4.2.1: A1 SKU support

In this release, BCM56965 A1 silicon has been validated. Device recognition and support for the BCM56965 A1 and BCM56967 A1 SKUs is available starting with SDK release 6.5.5.

Section 4.2.2: Hardware/Software Auto Negotiation update

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” was 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).

Hardware Auto Negotiation is supported on A1 devices only and is still under validation.

Section 4.3: 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 1.0.7 (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 4.3.1: 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 3 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.
FC Module support	Bringup	Host based management supported. I2C tunnel support for Modules connected to MT2 I2C. SFP: FC AN with gpio managed speed settings supported. QSFP: FC AN should be host managed.
LED	Bringup	Supported
Diagnostics	Beta	Register read/write, dsc dump, eye scan, PRBS supported. Phy diag supports multistage loopback. FW logging (dmpdbg) supported.
Traffic tests	Beta	TR19, TR72 supported.

Section 4.3.2: Notes and Known Limitations

- Tested with BCM956960, BCM56850, and BCM56760 SVK on PowerPC with Linux.
- Reset of port required after PRBS tests.
- FCoE COS to priority API not supported.
- Ethernet Repeater mode not supported

Section 4.4: BCM88470 (Qumran) family General Availability (GA) Release

This release is the GA version for the BCM88470 Family product line, following previously released 6.5.6 Beta version.
The subsequent sections describe the features validated for this release, known issues and bring-up guidelines.
For 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 4.4.1: Backward compatibility important notes

See also Section 4.5.1: BCM88670 Family “Backward compatible important notes” section.

Section 4.4.2: Validated Features

Basic data path, connectivity and Traffic Management features:

Interoperability with Jericho/Jericho+ devices in the same chassis

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 features
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
SLM/SLR by OAMP

Section 4.4.3: 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 Tags	SDK-92490	Native-VLAN-Editing: Bridging into VPLS tagged mode tunnel	8847X-AG302-R document, section Deeper Encapsulation Command Stack, example Bridging Into PWE Tagged Mode Tunnel with Service Delimiting Tags	6.5.5 (Done)
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	6.5.4 (Done)

			Editing Stage	
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 Editing Stage {VSI, Global-OutLIF.Profile.Domain)	--	--	8847X-AG302-R document, section Native Ethernet VLAN Editing Stage	Not planned
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 enhancements	8847X-AG302-R document, section MPLS Imposition Enhancements	6.5.4 (Done)
ITPP Network Headers Termination (Forwarding Copy)	--	None	8847X-AG302-R document, section, Ingress Transmit Packet Processor Enhancements	Not done yet
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)	88470-PG10X-R VSI Routing Interfaces section	6.5.4 (Done)
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 - see Errata section.

Association Identifier (MAID)			Enhancements	
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	--	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)	Will be documented in the next UM version (88470-PG1XX-R)	6.5.7 (Done) - see Errata section
OAM Hierarchical-LM per LIF (2 levels) – B0 only	SDK-75703	Hierarichical LM: Per LIF (New)	Will be documented in the next UM version (88470-PG1XX-R)	6.5.7 (Done) - see Errata section
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 (Done) - see Errata section
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 (Done)
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 (Done)
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 (Done)- see Errata section

Section 4.4.4: Major Bug fixes

ILKN:

In QAX, in order to allow loading ILKN port on more than one NBI block, the PLLs should have the same clock.

To enable the PLL binding, a new SoC property configuration is required:

serdes_nif_clk_binding_in0/1 = 1

suffix 0 refers to PML0 and suffix 1 to PML1.

When this feature is enabled, the input to PML0/1 will be forced from PMH PLL.

NOTE: when the new SoC property is in use, soc property serdes_nif_clk_freq_in0/1 value will be overridden!

Packet Processing:

See BCM88670-Family major bug-fixes section as well.

OAM: SLM/SLR are now supported by OAMP (true also for BCM88680-family)

OAM: Down MEP injection new format supports now also CCM piggyback

OAM: OAM over PWE injection with 2 OutLifs support CCM piggyback mode, LM ,DM.

Diag pp kbp last is now functional.

PON SAV is now functional.

Ingress-PMF Large direct lookup is now functional.

Section 4.4.5: Known issues

Network Interface:

LED is not functional.

Ingress latency measurement counting (by counter processor) isn't supported in this version.

Packet Processing:

See BCM88670 Known issues section.

OAM delay measurement using 1DM is not functional.

MAID48B is not functional due to CRC check failures

Down MEP egress injection new format and Hierarchical-LM Down-MEP by MD level will not work for LM, SLM cases since CCM packet will increment by mistake the LM counters.

OAM: OAM over PWE injection with 2 OutLifs and Hierarchical-LM by LIF will not work for SLM case since data packet will increment by mistake the LM counters.

Section 4.4.6: 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. From 6.5.6 version B0 registers changes included.

BCM88470 Family B0 revision is in the same quality as BCM88470 Family A0, 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 & Notes
Egress (ERPP) IP/MPLS Protocol Filters are Not Disabled For Control Packets	SDK-105095	8847X-ES105-R document EID#49	6.5.6 (Done)
NIF – ILKN Link Active Indication Not Correct For Small Ports	SDK-105097	8847X-ES105-R document EID#11	6.5.6 (Done)
Wrong forwarding code after MPLS Special Label Termination	SDK-105089	8847X-ES105-R document EID#4	6.5.7 (Done)
MPLS Preprocessing key construction does not take into account consecutive MPLS special labels	SDK-105090	8847X-ES105-R document EID#5	6.5.7 (Done)
MPLS Tunnel-Termination does not consider MTSE0 value in PHB and remarking	SDK-105092	8847X-ES105-R document EID#16	6.5.7 (Done)
IHP – UDP checksum of value 0 is marked as needed to be update	SDK-105091	8847X-ES105-R document EID#50	6.5.7 (Done)
ECMP LB Key0 and ECMP LB Key1 cannot be different when using configured LB logic	SDK-105087	8847X-ES105-R document EID#3	6.5.7 (Done)
PHP doesn’t consider BOS when deciding on next-protocol	SDK-105085	8847X-ES105-R document EID#2	6.5.7 (Done)
Different Location of IP Header Fields in Egress PMF for bridged packet and IP-MC Fallback-To-Bridge packets	SDK-105088	8847X-ES105-R document EID#1	6.5.8
Hierarchical LM	SDK-75703	8847X-ES105-R document EID#17	6.5.8

Section 4.4.7: New Features

See BCM88670-Family new features section, in addition

Basic data path, connectivity and Traffic Management features:

Packet Processing:

See BCM88670 New features section

Section 4.4.8: Important Notes

Configuring SerDes polarity in Kalia/MESH fabric links:
When configuring polarity on a fabric port, use the same SOC property as for non-MESH-QAX NIF port (i.e. with the suffix "phy" and not with "fabric" suffix). For example:

- phy_tx_polarity_flip_phy<X>
- phy_rx_polarity_flip_phy<X>

when X will be the physical port number (1-16).

Section 4.5: 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 6.5.6 are also supported in SDK 6.5.7.
The subsequent sections describe the increment in available features compared to 6.5.6, 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 4.5.1: Backward compatible important notes:

In case SDK upgrade from 6.4.X is required, please follow sections 4.5.1.1, 4.5.1.2, 4.5.1.3

In case SDK upgrade from 6.5.X which is not 6.5.6 is required, please follow sections 4.5.1.2, 4.5.1.3

In case SDK upgrade from 6.5.6 is required, please follow section 4.5.1.3

Section 4.5.1.1: SW compatibility guidelines 6.4.X to 6.5.X

Preserving SW interface 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 system, 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.

Section 4.5.1.2: SW compatibility guidelines 6.5.X (which is not 6.5.6) to 6.5.7

It is extremely important to read backward compatible important notes section over all SDK releases till 6.5.7. For example, in case upgrade from 6.5.4 to 6.5.7 is required, it is important to read backward compatible important notes section over SDK releases 6.5.5, 6.5.6, 6.5.7 (this document section 4.6.1.3).

Section 4.5.1.3: SW compatibility guidelines 6.5.6 to 6.5.7

Module	JIRA	Description	Devices affected
KBP/KAPS	--	SDK is aligned to KBP SDK 1.4.9	BCM88660-Family (only KBP) BCM88670-Family BCM88470-Family BCM88680-Family
KAPS DMA	SDK-111910	The soc property pmf_kaps_large_db_size should adhere to the following formula in Jericho: public_ip_frwrd_table_size + private_ip_frwrd_table_size + 5*pmf_kaps_large_db_size = 160000 The granularity is 10000 for each element, meaning pmf_kaps_large_db_size should be in multiples of 2000, each multiple of 2000 is equal to 8192 128bit PMF entries.	BCM88670-Family
MPLS/VPLS	SDK-106690	Entropy label (for VPLS, MPLS cases) and ELI (for MPLS cases) are now enforced to be BOS MPLS label in bcm_mpls_tunnel_initiator_create() and	BCM88670-Family

		bcm_mpls_tunnel_initiator_get(). In previous devices there was no such check and ELI/EL labels weren't added to packet which was misleading	
VPWS	SDK-108675	VPWS tagged mode PWE2AC direction: When using API bcm_flexible_cross_connect for same PWE-lif with both BCM_PORT_MATCH_PORT_VLAN_STACKED and BCM_PORT_MATCH_PORT_VLAN, native ETH double tagged packets will now looked up by 2 databases: single-VLAN and double-VLAN. In case both result will match: double tagged result will be prioritized. In previous versions, there was an assumption that double -VLAN and single-VLAN databases couldn't match on the same double-tagged packets.	BCM88670-Family BCM88470-Family BCM88680-Family
IPMC	SDK-110054 SDK-106027	bcm_ipmc_add: When REPLACE flag is used to update an IPMC entry that doesn't exist, it now returns BCM_E_NOT_FOUND. In previous release it created a new entry. The behavior is changed for both IPv4 & IPv6 entries.	BCM88660-Family BCM88670-Family BCM88470-Family BCM88680-Family
IPMC	SDK-108944	IMPORTANT NOTE: If the DIP that was provided to the IPMC APIs (add, find remove) is not a MC one, regardless of the mask and device, an error will be returned. In previous releases, non-MC addresses were accepted. The behavior is changed for both IPv4 & IPv6 entries.	BCM88660-Family BCM88670-Family BCM88470-Family BCM88680-Family
IPMC	SDK-111142	Added check for IPMC support valid (according to bcm_ipmc_enable behavior) in the beginning of the APIs that didn't have one.	BCM88660-Family BCM88670-Family BCM88470-Family BCM88680-Family
ILKN	SDK-109060	ILKN 1/3/5 Lane swap configuration: change is not compatible with SDK-6.5.5 and SDK-6.5.6. Compatible with prior versions. When working with ILKN port, it is always recommended to use ILKN lane map SOC property. However, a bug was discovered in 6.5.5 (which was also included in 6.4.11 SP), which caused a change in the ILKN lane map SOC property calculation. One outcome of the bug was that the expected SOC property value in 6.5.5/6 is different than the expected SOC property value in previous versions (only for ILKN1/3/5). This bug is now fixed. With the fix, the expected value for ILKN lane map SOC property is the same as the value in versions prior to	BCM88670-Family BCM88470-Family BCM88680-Family

		6.5.5.	
OAM	SDK-111746	Changes to default up MEP snoop command to CPU (used for LTM packets). packets will be snooped in the same way but the actual internal command will be trap. This to allow original outlif to appear on the system headers. Additionally, the calling sequence for setting packet actions to snoop has been changed - there is no longer a need to define a snoop command along with the snoop. For a detailed account, refer to mip_set_snoop_by_opcode() in cint_oam.c	BCM88660-Family BCM88670-Family BCM88470-Family BCM88680-Family
EEDB	SDK-110846	EEDB default entry value is now changed. In previous releases, EEDB use_as_data_entry is set to 1. On some cases, when packet hits EVPN program, use_as_data_entry is parsed as outlif. Therefore empty EEDB entry got outlif 1 when parsing by EVPN program. This bug is now fixed. use_as_data_entry is set to 0 by default.	BCM88660-Family BCM88670-Family BCM88470-Family BCM88680-Family
Load-balancing	SDK-110443	ELI search was disabled as it wasn't working as expected, so when using bcm_switch_control_set with the bcmSwitchHashELISearch option an unavailable error will be returned. A properly working ELI implementation will be added under new LB scheme only in version 6.5.8	BCM88670-Family BCM88470-Family BCM88680-Family
OAM	SDK-109424	Disable MEP sending OAM packets behavior changed. An endpoint may be create with ccm_period set to one of the 8 defined values. If the endpoint will then be updated with ccm_period set to 0 then the endpoint will still accept CCMs with initial period, however CCMs will no longer be transmitted. The soc property custom_feature_oam_ccm_rx_wo_tx Is now deprecated.	BCM88470-Family
OAM	SDK-108532	My-CFM-MAC l2 station API changes for OAM: bcm_l2_station_add API will now return an error when a multicast mac is passed in the bcm_l2_station_t.dst_mac parameter of the API with flags parameter set to BCM_L2_STATION_OAM.	BCM88660-Family BCM88670-Family BCM88470-Family BCM88680-Family
ERSPAN	SDK-101456	ERSPAN tunnel doesn’t use anymore bcmSwitchL2GreProtocolType to configure GRE.Next-protocol. This reduce the conflict resource of GRE.Unknown-next-header with other applications (like MPLS over GRE)	BCM88660-Family BCM88670-Family BCM88470-Family BCM88680-Family

Field	SDK-106218	IMPORTANT NOTE: When setting SOC property field_presel_mgmt_advance_mode, it is now prohibited to create presel objects using the bcm_field_presel_create_id API, and using this API will result in failure and an error message. For this advanced mode, the following APIs should be used: bcm_field_presel_create_stage_id (unit, stage, presel_id), presel_id will later be used with flag BCM_FIELD_QUALIFY_PRESEL_ADVANCED_MODE_STAGE_xxx. Please use the following for reference: cint_field_presel_create_stage_id.c cint_field_presel_mgmt_advanced_mode.c	BCM88660-Family BCM88670-Family BCM88470-Family BCM88680-Family
DRAM	SDK-105459	SOC property ext_ram_total_size should no longer be rounded down to the closest thousand but should instead represent exactly the amount of Dram on the system. This JIRA fixes risks for Dram CRCs in certain configurations. As a result of this fix usable Dram size can be reduced by up to ~138MB - failing to update the SOC property will result in losing also the difference between the actual Dram size to value inserted in the SOC property after the user manually rounded it down.	BCM88670-Family
OAM	SDK-112345	OAMP TX machine pads now all generated packets to 44 bytes(not include system header size). This means that all packets out of the OAMP and at the start of the IRE will have at least 44 bytes + System-header bytes. This is different than before, where the OAMP didn't have any minimal packet size and could cause NIF drop issues. Because of that, expect now that some of the packet flows from OAMP will have different packet-sizes than before.	BCM88670-Family BCM88470-Family BCM88680-Family
CGM		Default CGM thresholds were modified to values better fit for most scenarios (both at ingress and at egress TM). This affects CGM configuration after initialization, unless overridden by specific API calls.	BCM88470-Family

Section 4.5.2: New Features

Packet Processing:

KBP: SDK-104900 It is now possible to use external forwarding result 24b size (instead of 48b) when device is connected to NL12K. Reducing the size allow more room for ACL

result size on the expense of supporting Host entries format in KBP. True also for BCM88680-family, BCM88470-family.
KBP: SDK-105764 IPv6 ACL can now include DIP and SIP qualifiers even when IPv6 forwarding is not done in the external device (KBP). True also for BCM88680-family, BCM88470-family.

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.

Section 4.5.3: 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.

Packet Processing:

L2: SDK-106576 On some cases when adding large number of LEM entries, LEM shadow was not in sync (relevant only when LEM shadow is enabled). True for BCM88660, BCM88470, BCM88680 families as well.

VRRP: SDK-107060, SDK-107155, SDK-107263, SDK-107276 - Fixed few VRRP issues, major one VRRP IPv6 was not functional. see JIRAs detailed description. True for BCM88470, BCM88680 families as well.

Tunnel IPv6: SDK-107530 bcm_tunnel_initiator_clear does not delete global_to_local LIF mapping in case of IPv6 tunnel. Issue is now fixed. True for BCM88470, BCM88680 families as well.

MIM: SDK-107983 bcm_mim_port_delete now correctly delete ports associated with the second core. True for BCM88680 family as well.

BFD: SDK-111243 BFD Tunnel replace (bcm_tunnel_initiator_create with replace flag) failed when SOC-property bfd_extended_ipv4_src_ip was enabled. This issue is now fixed. True for BCM88660, BCM88470, BCM88680 families as well.

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.

Section 4.5.4: Errata

The list below relates to major open bugs that are not resolved:
Basic data path, connectivity and Traffic Management features:

Packet Processing:

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

SDK-112662 BFD single-hop can't be enabled in case UDP tunnel (soc property udp_tunnel_enable) or VXLAN (bcm886xx_vxlan_enable) are enabled .It is mutually exclusive. Also true for BCM88470, BCM88680 families.

Load-balancing BOS-search enabled (bcm_switch_control_set , enum: bcmSwitchTrunkHashMPLSLabelBOS) is not functional. Shouldn’t be used.

Section 4.5.5: Important Notes

The following APIs may not be called under traffic:

- bcm_rx_start/stop (BCM88770).
- bcm_fabric_link_topology_set (BCM88675, BCM88680, BCM88470).
- the shell command "fabric force..." (BCM88675, BCM88680, BCM88476).

In order to disable the traffic, use the following APIs:
- FE device: bcm_fabric_control_set with bcmFabricIsolate/bcmFabricShutdown parameter.
- FAP device:
1. bcm_stk_module_enable
2. bcm_fabric_control_set with bcmFabricControlCellsEnable parameter.

In case these APIs are called under traffic, they will not return an error indication rather will issue a warning message, according to the configured BSL logging level.
Changing fabric port speed of an FE/FAP device:
Scope:
Fabric SerDeses are grouped in quads, sharing a single clock. Due to this fact, changing a speed of a single link may impact other links in the same quad. This document describes the proper sequence that must be applied when changing a fabric port (link) speed. Changing speed not according to these guidelines may lead to undesired device behavior.

Guidelines:
Whenever setting fabric port speed, in case the original speed and the new speed are not in the same row in Table 1 - Fabric PHY Rates, follow the sequence below:
1. Isolate all ports in the quad: bcm_fabric_link_control_set with the flag bcmFabricLinkIsolate.
2. Disable all ports in the quad: bcm_port_enable_set.
3. Set speed for all ports in the quad: bcm_port_speed_set.
4. Enable all ports in the quad if originally enabled: bcm_port_enable_set.
5. De-isolate all ports in the quad if originally not isolated: bcm_fabric_link_control_set with the flag bcmFabricLinkIsolate.

Note: isolation/de-isolation is necessary to avoid traffic loss.

Ref Clk(MHz)	Full rate	Half rate	Other
156.25	20625	10312	
156.25	21875	10937	
156.25	25000	12500	6250
156.25	25781		
125	20625	10312	
125	22500	11250	
125	23000	11500	5750
125	25000	12500	6250

Section 4.5.6: Connectivity with KBP Optimus Prime (OP)

In 6.5.7, we introduce the capability to connect BCM88670-Family to KBP Optimus Prime (new KBP device). Note that KBPSDK version must be at least 1.4.9.

Bring-up is done, basic functionality works. Please approach AE team for bring-up guidelines.

Following known issues exist:

- There are ROP instability when bringing up the system.
- Dynamic results size is not supported.
  The fixed results sizes of the 6 lookups are: 48b, 24b, 24b, 16b, 32b, 32b.
  Soc properties definition should be as follows:
  config add ext_tcam_result_size_segment_0=48
  config add ext_tcam_result_size_segment_1=24
  config add ext_tcam_result_size_segment_2=24
  config add ext_tcam_result_size_segment_3=16
  config add ext_tcam_result_size_segment_4=32
  config add ext_tcam_result_size_segment_5=32
- IPv4/6 MC are not functional
- ACL is not functional

Section 4.6: BCM88680 (Jericho+) Family GA Release

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

Section 4.6.1: Backward compatible important notes

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

Section 4.6.2: 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 4.6.4: New Features

Packet Processing:

See BCM88670-Family, BCM88470-Family New features sections

Section 4.6.5: Major Bug fixes

Packet Processing:

See BCM88670, BCM88470 families Major Bug fixes section

Section 4.6.6: Known issues

Packet Processing:

See BCM88670, BCM88470 families Known issues section

SDK-108420 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.

No Warmboot support for Ingress-PMF Large direct lookup (KAPS DMA).

Section 4.7: 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.

Section 4.7.1: Important Notes

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

Section 4.7.2: Major Bugfixes

- None

Section 4.8: BCM88270 (Qumran-UX) Family Beta Release

This release is a beta version for the BCM88270-Family product line, following previously released 6.5.6 Beta 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 4.8.1: Backward compatible important notes

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

Section 4.8.2: Validated Features

Basic data path, connectivity and Traffic Management features:

Register & Memory access including DMA
Supported SKUs:
Qumran-UX: 88270, 88272, 88273, 88278
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
Ingress and Egress compensation
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
Basic Link Bonding features

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 features
TRILL
L2GRE
802.1BR
ROO, ROO VPLS, ROO VXLAN
EVPN
VPLS protection - 2 levels
MIM/SPB
Support for additional VLAN tag TPIDs
3 Ethernet VLAN tags parsing
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 Automatic RDI assertion

Section 4.8.3: New Features

Packet Processing:

Section 4.8.4: Major Bug fixes

Packet Processing:

Section 4.8.5: 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
DRAM/VSQ/LED/FC/ECN/CRPS/SyncE are not functional.
Link Bonding control message is not functional.

Packet Processing:

OAM SLM/SLR by OAMP is not functional.
UP MEP is not functional when UP MEP id is larger than 255.

Counter Processor:

Counting based on PMF counter-pointer is not functional.

Section 4.9: BCM88660 (ARAD+), BCM88650 (ARAD) Release

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

Section 4.9.1: Important notes

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

Section 4.9.2: 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.

SER LEM: SDK-111038 In soft error recovery (SER) for exact match memories, recovery procedure may have failed in certain cases, when global LEM counter had to be corrected. An issue in access level was fixed to allow proper recovery procedure.

Section 4.9.3: Errata

None

Section 4.10 Updated External PHY Support

Section 4.10.1 BCM84888 (Orca) support

The Broadcom BCM84888 is a quad 10GBASE-T/5GBASE-T/2.5GBASE-T/100BASE-T/100BASE-TX Ethernet CMOS transceiver. In this release, BCM84888 is supported with GA level quality. The PHY features have been tested with the BCM56860 switch device.

Current device limitations in this release:

1588 is not yet supported.
PHY loopback for 5G is not supported.

Section 4.10.2 BCM82332 (Dino) support

In this release, warmboot support has been added and validated. Note that features have been tested with the BCM56860 switch device.

Current device limitations in this release:

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
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 4.10.3 BCM54190 support

Preview support has been added for the BCM51490 external PHY driver in this release. Please see Section 6 of the SDK 6.5.6 release notes for details about APIs added.

Section 4.10.4: Interoperability testing for new PHY/switch combinations

The following switch and PHY combinations have been interoperability tested in the SDK 6.5.7 release. Below lists only the features that have been tested and supported in this release.

Section 4.10.4.1: BCM56860 and PHY BCM82322

Features supported:

- 10G Forced Speed and Auto-Negotiation modes
- 1G supported in Forced Speed mode only
- Diagnostics
  - PRBS polynomials 4/5/6, PCS/PMD loopback, TX Squelch are not supported
- Limitations
  - 1G Auto-Negotiation, 40G not supported in this release

Section 4.10.4.2: BCM56160 and PHY BCM82780

Features supported:

- 10G Forced Speed and Auto-Negotiation modes
- Diagnostics
  - PRBS, DSC dump, Eyescan and TX Squelch are supported
- Limitations
  - PCS/PMD Loopback not supported in this release
  - Port disable/enable in Auto-Negotiation mode not supported in this release

Section 4.10.4.3: BCM56565 and PHY BCM84756

Features supported:

- 10G Forced Speed mode only
- 1G Forced Speed and Auto-Negotiation mode supported
- Diagnostics
  - Line side PRBS, DSC dump, PCS/PMD loopback and Eyescan are not supported
- Limitations
  - 1G PRBS not supported in this release

Section 4.10.4.4: BCM56760 and PHY BCM84858

Features supported:

- 100M/1G/10G Auto-Negotiation mode supported
- Forced Speed mode applicable to 100M only
- PCS Loopback
- Diagnostics
  - PRBS (line side), DSC dump, Eyescan and TX Squelch are not supported
  - System side PRBS and DSC dump supported

Section 4.10.4.5: BCM56760 and PHY BCM84888

Features supported:

- 10G/5G/2.5G/1G/100M Auto-Negotiation mode supported
- Forced Speed mode supported for 100M only
- Diagnostics
  - Line side PRBS, DSC dump, Eyescan, PMD Loopback, TX Squelch not supported
  - PCS Loopback supported
  - System side PRBS and DSC dump supported

Section 4.10.4.6: BCM88670 and PHY BCM82780

Features supported:

- 1G, 40G Forced Speed mode
- 10G Forced Speed and Auto-Negotiation modes
- Diagnostics
  - PRBS, DSC dump, Eyescan and TX Squelch are supported
- Limitations
  - Port disable/enable in Auto-Negotiation mode 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: PSIRT Reporting Process

Broadcom treats security vulnerability issues reported by customer Product Security Incident Response Teams (PSIRT) with very high importance and urgency. Please ensure that any such issues reported and filed by your organization through the Broadcom customer support portal specifically use the acronym “PSIRT” in the CSP case summary and/or description. This will allow the Broadcom engineering teams to track, analyze, and address these issues as quickly as possible.

Section 5.2: 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 5.3: 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 5.4: ALPM enhancements for BCM56960, BCM56860, and BCM56850

Starting in 6.5.6, 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, BCM56960 and BCM56970, the ALPM utilization has been improved through introducing bucket merge and bucket repartition. As a result, the guaranteed minimum route capacity for IPv4 and IPv6 in ALPM mode is now increased.

Section 5.5: Interface related changes to “ps” command from SDK 6.5.6

Starting in 6.5.6, enhancements have been made to have a consistent behavior when change in interfaces occur. Previously, Speed IDs stored in the register was mapped to multiple user interface types.

For example, when user enters SR4, CR4, KR4 or XLAUI interface for a 40G port, the actual speed ID stored in the register is *SPEED_40G_KR4.* When bcm shell command “ps” is used to display interfaces it return unexpected/inaccurate interface types because cannot recover the user-entered interface type precisely by using the speed ID “as is” from the register.

In the 6.5.5 and earlier releases, **errors were not always flagged** when the user enters an invalid interface type based on the port speed and encap. In 6.5.6 onwards, **errors are always flagged** when the user enters an invalid interface type.

To change the interface type of a port without modifying the current port speed and encap is a two-step process:

1. Specify interface type first so that the new interface type is stored in the software database
2. Then specify a speed to force the interface type stored in the software database to be written into the hardware register

[Example} if we have a 10G SFI port, to make it 10G CR, steps needed are:

```
BCM.0> port xe0 if=cr
```

CR is stored in the software database

```
BCM.0> port xe0 sp=10000
```

CR is fetched from the software database and written into the hardware register

```
BCM.0> ps
```

Shows 10G/CR in both 6.5.5 and 6.5.6 after previous step

Section 5.5.1: Expected Behavior Changes

Section 5.5.1.1: Speed set is allowed only for valid interfaces

6.5.6 checks the interface type in the first step. User cannot proceed to the second step if the provided interface type is not valid according to the current speed and encap.

[Example] if we have a 10G SFI port, and user types in an invalid new interface type such as CAUI, which is only valid for 100G port:

SDK 6.5.5 and before

```
BCM.0> port xe0 if= CAUI
```

Software database is updated to CAUI

```
BCM.0> port xe0 sp=10000
```

If the port is connected to an external PHY, user will see error returned at this step

`BCM.0> ps` shows 10G/SFI after this step

If the port is not connected to an external PHY, the internal PHY driver will silently change CAUI into the default interface type XFI based on speed 10G

`BCM.0>` ps shows 10G/XFI after this step

In either case the software database is updated incorrectly and may cause error returned from follow on operations.

SDK 6.5.6 onwards

```
BCM.0> port xe0 if= CAUI
```

PORTMOD returns error because CAUI is not a valid interface type based on the current port speed 10G

Software database remains untouched

`BCM.0> ps` shows 10G/SFI after this step

Section 5.5.1.2: Default Port speed

Whenever user changes the port speed, 6.5.6 SDK changes the interface type to a default value based on the new port speed

[Example] User wants to change a 10G SFI port to 1G GMII

6.5.5 and before

```
BCM.0> port xe0 if=gmii
```

```
BCM.0> port xe0 sp=1000
```

`BCM.0> ps` shows 1G/GMII after this step

6.5.6 onwards

```
BCM.0> port xe0 sp=1000
```

After this step, the interface type is changed to the default value SGMII based on the new speed 1G

To change the interface type from SGMII to GMII, we need to use the two-step process below

`BCM.0> ps` shows 1G/SGMII after this step

```
BCM.0> port ge0 if=gmii
```

```
BCM.0> port ge0 sp=1000
```

`BCM.0> ps` shows 1G/GMII after this step

Section 5.5.1.3: Encap change will be considered to set a default interface

Whenever user changes the port encap, SDK 6.5.6 onwards changes the interface type to a default value based on the new port encap

[Example] User wants to change a 40G IEEE XLAUI port to 42G HG2 CR4

6.5.5 and before

```
BCM.0> port xe0 encap=hg2
```

```
BCM.0> port hg0 if=CR4
```

```
BCM.0> port hg0 sp=42000
```

`BCM.0> ps` shows HG/42G/CR4 after this step

6.5.6 onwards

```
BCM.0> port xe0 encap=hg2
```

After this step, port xe0 becomes 40G HG2 with default interface type KR4

`BCM.0> ps` shows HG/40G/KR4 after this step

```
BCM.0> port hg0 sp=42000
```

After this step, port xe0 becomes 42G HG2 with default interface type KR4

To change interface type to CR4, user has to follow the two-step process below. Note that every time when user changes speed or encap, the interface type will be restored to its default value based on the new speed or encap.

`BCM.0> ps` shows HG/42G/KR4 after this step

```
BCM.0> port hg0 if=cr4

BCM.0> port hg0 sp=42000
```

`BCM.0> ps` shows HG/42G/CR4 after this step

Section 5.5.1.4: Interface type accuracy

User sees accurate interface type through “BCM.0> ps” after 6.5.6

[Example] User changes a 40G IEEE XLAUI port it to 40G CR4

6.5.5 and before

“BCM.0> ps” may show KR4 because CR4 and KR4 share the same speed ID

Similar problem happens to LR4/SR4, CR/KR and many other interface types

6.5.6 onwards

User sees accurate interface type CR4 even after warmboot

Section 5.6: SDK releases out of active engineering support

The following releases are out of active engineering support:

SDK 6.4.x releases: 6.4.9, 6.4.8, 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

Customers are recommended to use this release for new product development.

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***
BCM56260 family	152	36
BCM56860 family	304	71
BCM56960 family	231	67
BCM56850 family	196	38

BCM56840_PLUS	36	3
BCM56340 family	43	12
BCM56640 family	81	17
BCM56450 family	102	34
BCM56150 family	11	2
BCM56440 family	38	19
BCM53440 family	30	5

Section 5.7: Alert on future support deprecation for older devices

Per Broadcom policy, as older devices are discontinued due to end of life (EOL), their SW support is also deprecated in SDK releases beyond the device EOL date.

Section 5.8: Warmboot Notes and Considerations

Section 5.8.1: Validated Warmboot upgrades

Warmboot upgrades in the table below have been validated on the SDK switch in this release.

Software Upgrade	***Supported***
6.4.11 to 6.5.7	Yes
6.5.6 to 6.5.7	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
BCM56970 - like-to-like testing only

Warmboot testing is not performed with PHY devices attached. Please note that Broadcom will no longer perform upgrade testing from 6.4.x train after this release.

Section 5.8.2: Validated Warmboot downgrades

Broadcom has performed limited release downgrade testing from 6.5.7 to 6.5.6 and 6.5.7 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 5.8.3: 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 5.8.4: Warmboot upgrade/downgrade considerations

There may be unrecoverable errors when performing warmboot upgrade from an SDK release doesn't support BCM56970 to an SDK release such as SDK-6.5.7, due to added support for BCM56970. This is due to the bookkeeping information in the newer SDK version will be recovered from the scache buffer recreated based on the external storage data saved by previous SDK versions. In some specific cases such as SDK-111724, as the macro value in the legacy version can be known, it is possible to provide a temporary patch by examining all the modules using SOC_MAX_NUM_PORTS in the warmboot bookkeeping information definition, and take it into consideration during warmboot recovery.

If performing a warmboot downgrade from SDK 6.5.7 to SDK 6.5.6, a patch from SDK-105891 may be required to address a mirror destination ID failure. In SDK 6.5.6, the warmboot recovery for mirror ingress/egress parameters was not employed and the patch applied to SDK 6.5.6 before performing the downgrade will be required.

Section 5.8.5: 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 5.9: Release notes for ILKN interface in 88675, 88635, 88680 and 88470 families

The below list refers to release notes for the ILKN in the DNX 88675, 88635, 88680 and 88470 products.

1. **New SOC property for solving ILKN first packet drop issue**: A new SOC property, *ilkn_first_packet_sw_bypass*, was added to solve the ILKN first packet drop issue. Setting this SOC property to 1 will enable the SW solution by delaying the ILKN Rx enable only to after the phy connection is set between both peers. Note that the use of this SOC property requires the user to activate SW link scan on the same ILKN port!
2. **ILKN 1/3/5 Lane swap configuration:** The logic of using the *ilkn_lane_map* SOC property for odd ILKN ports (i.e. ILKN1, ILKN3 or ILKN5) was wrongly modified in 6.4.11 SP, 6.5.5 and 6.5.6 versions so it was not compatible with the logic of the even ports (ILKN0, ILKN2 or ILKN4). In 6.5.7, the code was reverted to use the original logic for the odd ports, making it compatible with the logic of the even ILKN ports as well as with the logic of the odd ILKN ports in versions that were prior to 6.5.5 and 6.4.11 SP. Note that compared to 6.5.5, 6.5.6 and 6.4.11 SP releases this change is not backward compatible and may require changes in customer SOC property setup.
3. **Better validity checking of ILKN allocated segments:** ILKN segments are a resource shared between the two ILKN ports on the same ILKN core. The number of segments required for a port operation is determined by the number of lanes, serdes rate, short_burst value and the device core clock. A new logic is introduced to determine the number of required segments according to the four factors above. If there are not enough available segments for the ILKN port, an error will be returned (can occur during init or when calling dynamic port API). The number of required segments is highly dependant on the value of ILKN short burst, controlled by the *ilkn_burst_short* SOC property. Typically this value should be set to 64B (although if not specified, default is

32B). In case of hitting the out-of-segments error condition, please check the value of *ilkn_burst_short*. Increasing this value is likely to resolve the problem.

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-PG657-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.x/RELDOCS directory in the release package.

Section 6.1: Class of Service Queue Configuration

Table 6.1: Port Bandwidth Flags

Flag	Description	
BCM_COSQ_BW_EAV_TAS_MODE	Avoid burstiness when gate is opening or after queue is empty	New
BCM_COSQ_BW_ALT_CALC	Use alternative building algorithm for bandwidth calculation	New
BCM_COSQ_BW_TAS_TOKEN_FREEZE	Freeze shaper token when gate is closed to avoid burstiness	New

Table 6.2: COSQ Event Collection Macros

Macro	Description	
BCM_COSQ_EVENT_TYPE_GET(_event_types, _event_type)	Get the status of the specified event type in the collection?	New
BCM_COSQ_EVENT_TYPE_CLEAR(_event_types, _event_type)	Clear the specified event type in the collection	New
BCM_COSQ_EVENT_TYPE_CLEAR_ALL(_event_types)	Clear all event types in the collection	New
BCM_COSQ_EVENT_TYPE_SET_ALL(_event_types)	Set all event types in the collection?	New
BCM_COSQ_EVENT_TYPE_SET(_event_types, _event_type)	Set the specified event type in the collection	New

Table 6.3: bcm_cosq_stat_t

Name	Description	
bcmCosqStatTASTxOverrunPackets	Number of transmission overrun events where a packet is still being transmitted at the time when the transmission gate of the queue closed.	New

Table 6.4: Parameter for bcmCosqControlAlternateStoreForward

Name	Description
bcmCosqAsfModeDisabled	No cut-through, i.e. store-and-forward mode.
bcmCosqAsfModeStoreAndForward	Store-and-forward mode.
bcmCosqAsfModeAlternateStoreForward	Alternate-store-forward (cut-through) mode.

Table 6.5: CoSQ Control Type Values

Value			
bcmCosqControlAlternateStoreForward	Configure ASF mode.	Refer to bcmCosqControlAlternateStoreForward Parameter table (unknown XREF bcm_cosq_asf_mode_t) for a detailed description of modes.	New

The followings are used for TAS profile management APIs.

```
typedef int bcm_cosq_tas_profile_id_t ;

#define MAX_COSQ_TAS_CALENDAR_TABLE_SIZE    (128)

typedef struct bcm_cosq_tas_profile_s {
    int num_entries;   /* Specify the number of entries in the calendar table. */
    bcm_cosq_tas_entry_t entries[MAX_COSQ_TAS_CALENDAR_TABLE_SIZE];    /* Specify
the interval, gate states, attribute entry sets in the calendar table. */
    bcm_ptp_timestamp_t ptp_base_time;     /* Specify base time in PTP time*/
    uint32  ptp_cycle_time; /* Specify cycle time in ns */
    uint32? ptp_max_cycle_extension; /* Specify max cycle extension time in ns.
```

```
Maximum time allowed for a cycle to be extended before profile change. */
} bcm_cosq_tas_profile_t;

typedef struct bcm_cosq_tas_entry_s {
    uint32  ticks;  /* The scheduling interval in ticks. The tick interval is
defined in bcmCosqTASControlGateControlTickInterval. */
    uint32  state;  /* Specify gate state for each traffic class queue  in
bitmap. (1 bit per queue) */
    uint32  flags;  /*  Defined in BCM_COSQ_TAS_ENTRY_F_XXX */
}bcm_cosq_tas_entry_t;

#define BCM_COSQ_TAS_ENTRY_F_PREEMPT 0x00000001  /* Request to hold the preempt
traffic */

/* The encoded value used to assigned in ticks field on bcm_cosq_tas_entry_t */
#define BCM_COSQ_TAS_ENTRY_TICKS_STAY (-1)  /* Specify to stay in this entry
forever in non PTP_MODE or until the next cycle in PTP_MODE. */
#define BCM_COSQ_TAS_ENTRY_TICKS_GO_FIRST (0)  /* Specify to jump to the first
entry. The state of this entry will be ignored. */


typedef bcm_cosq_tas_profile_status_s {
    bcm_cosq_tas_profile_state_t profile_state; /* profile state */
    bcm_ptp_timestamp_t config_change_time; /* The actual config change time.*/
    int num_entries; /* The actual number of entries in hardware calendar table.
*/
    bcm_cosq_tas_entry_t entries[MAX_COSQ_TAS_CALENDAR_TABLE_SIZE]; /* The actual
entries in hardware calendar table */
} bcm_cosq_tas_profile_status_t;


/* TAS profile state */
typedef enum bcm_cosq_tas_profile_state_e {
    bcmCosqTASProfileInit = 0,      /* the state after profile created. */
    bcmCosqTASProfileCommitted = 1, /* the state after profile committed but not
                                       yet written to HW table */
    bcmCosqTASProfilePending = 2,   /*  the state after profile is set to
                                       hardware but not yet effective */
    bcmCosqTASProfileActive = 3,    /* the state indicate the profile is
                                       effective */
    bcmCosqTASProfileExpired = 4,   /* The profile is no longer effective. */
    bcmCosqTASProfileError = 5,     /* the state indicate the profile is error
                                       when any error event happens */
    bcmCosTASProfileCount = 6       /* This should be the last one. */
} bcm_cosq_tas_profile_state_t;
```

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 Parameter for bcmCosqControlAlternateStoreForward

```
typedef enum bcm_cosq_asf_mode_e {
    bcmCosqAsfModeDisabled = 0,         /* No cut-through, i.e.
                                           store-and-forward mode */
    bcmCosqAsfModeStoreAndForward = bcmCosqAsfModeDisabled, /* Store-and-forward
mode */
    bcmCosqAsfModeAlternateStoreForward = 1, /* Alternate-store-forward
(cut-through)
                                           mode */
    bcmCosqAsfModeCount = 2             /* Must be last. Not a usable value */
} bcm_cosq_asf_mode_t;
```

New Cosq event type

```
typedef enum bcm_cosq_event_type_e {
    bcmCosqEventTASProfileChange = 0,   /* TAS: the profile change is completed.
*/
    bcmCosqEventTASECCErr = 1,          /* TAS: uncorrectable ECC error in
                                           profile calendar table. */
    bcmCosqEventTASNextCycleTimeErr = 2, /* TAS: next cycle start time has
                                           passed. */
    bcmCosqEventTASBaseTimeErr = 3,     /* TAS: base time for the new profile
                                           has passed. */
    bcmCosqEventTASProfilePtrErr = 4,   /* TAS: index to the profile calendar
                                           table is overrun. */
    bcmCosqEventTASPTPOutOfSyncErr = 5, /* TAS: IEEE1588 PTP time is out of
```

```
                                                sync. */
    bcmCosqEventCount = 6                    /* This should be the last one. */
} bcm_cosq_event_type_t;
```

Table 6.6: BCM CoSQ PFC Configuration

Parameter	Description	
xoff_fadt_threshold	Assert backpressure when number of resources used is at or above this FADT threshold	New
xon_fadt_offset	Remove backpressure when number of resources used drops below FADT threshold defined by xoff_fadt_threshold - fadt_offset	New

bcm_cosq_bst_stat_extended_clear

Clear the current statistic/count of specified BST profile

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_bst_stat_extended_clear(int unit, bcm_cosq_object_id_t *id,
                                     bcm_bst_stat_id_t bid);
```

PARAMETERS

unit	BCM device number
id	Compound index containing gport, cosq, buffer, etc. See (unknown XREF bcm_cosq_object_id_t_init) .
bid	BST stat ID to identify the COSQ resource/object

DESCRIPTION

Same as (unknown XREF bcm_cosq_bst_stat_clear) , with the difference that BST statistic can be cleared on more basis than gport/cosq.

RETURNS

BCM_E_NONE

BCM_E_XXX

bcm_cosq_bst_stat_extended_get bcm_cosq_bst_stat_extended_get32

Get the current statistic/count of specified BST profile

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_bst_stat_extended_get(int unit, bcm_cosq_object_id_t *id,
                                   bcm_bst_stat_id_t bid, uint32 options, uint64
*pvalue);
int bcm_cosq_bst_stat_extended_get32(int unit, bcm_cosq_object_id_t *id,
                                     bcm_bst_stat_id_t bid, uint32 options,
uint32 *pvalue);
```

PARAMETERS

unit	BCM device number
id	Compound index containing gport, cosq, buffer, etc. See (unknown XREF bcm_cosq_object_id_t_init) .
bid	BST stat ID to identify the COSQ resource/object
options	options to perform clear-on-read
pvalue	stat value to return.

DESCRIPTION

Same as (unknown XREF bcm_cosq_bst_stat_get) and (unknown XREF bcm_cosq_bst_stat_get32) , with the difference that BST statistic can be retrieved on more basis than gport/cosq.

RETURNS

BCM_E_NONE

BCM_E_XXX

bcm_cosq_event_register bcm_cosq_event_unregister

Register/unregister a callback to handle the Cosq events

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_event_register(
    int unit,
    bcm_cosq_event_types_t event_types,
    bcm_gport_t gport,
    bcm_cos_queue_t cosq,
    bcm_cosq_event_cb cb,
    void *user_data)
int bcm_cosq_event_unregister(
    int unit,
    bcm_cosq_event_types_t event_types,
    bcm_gport_t gport,
    bcm_cos_queue_t cosq,
    bcm_cosq_event_cb cb)
```

PARAMETERS

unit	BCM device number
event_type	(IN) The set of COSQ events for which the specified callback should be called
gport	(IN) gport id
cosq	(IN) cos queue
cb	(IN) A pointer to the callback function to call for the specified COSQ events

user_data (IN) (for "_register") Pointer to user data to supply in the callback

DESCRIPTION

The call back function registration by specifying port or traffic class to handle the COSQ events. If the event is not on port or traffic class basis, the gport and cosq value should be -1. Or user do not want to register the event on specified port or queue basis, the gport or cosq value should assign to -1.

RETURNS

BCM_E_NONE

BCM_E_XXX

---

bcm_cosq_tas_control_set bcm_cosq_tas_control_get

Set/get TAS control configurations.

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tas_control_set(
    int unit,
    bcm_gport_t gport,
    bcm_cosq_tas_control_t type,
    uint32 arg)
int bcm_cosq_tas_control_get(
    int unit,
    bcm_gport_t gport,
    bcm_cosq_tas_control_t type,
    uint32 *arg)
```

PARAMETERS

unit BCM device number

gport	(IN) gport id
type	(IN) tas control type defined in bcm_cosq_tas_control_t
arg	(IN) (for "_set") tas control value.
arg	(OUT)(for "_get") tas control value.

DESCRIPTION

Set/get various configurations for TAS. Most of the TAS configurations are controlled on a gport basis. For the particular configuration on system basis, the gport should assign to -1.

TAS Control Type

```
typedef enum bcm_cosq_tas_control_e {
    bcmCosqTASControlTASEnable = 0,     /* Enable the TAS. */
    bcmCosqTASControlUsePTPTime = 1,    /* PTP time is used for TAS synchronous
                                           operation. */
    bcmCosqTASControlInitGateState = 2, /* Gate state when TAS is disabled. Or
                                           the init gate state in the *Restart*
                                           process. */
    bcmCosqTASControlErrGateState = 3,  /* Gate state when error condition
                                           occurs during an old cycle is
                                           running. */
    bcmCosqTASControlGatePurgeEnable = 4, /* Purge express packet when gate is
                                           closed. */
    bcmCosqTASControlGateControlTickInterval = 5, /* Gate control ticks interval in ns.
*/
    bcmCosqTASControlTASPTPLock = 6,    /* Indicate the PTP time is locked or
                                           out of sync. This type is on system
                                           basis. */
    bcmCosqTASCountrolCount = 7         /* This should be the last one. */
} bcm_cosq_tas_control_t;
```

RETURNS

BCM_E_NONE

BCM_E_XXX

bcm_cosq_tas_profile_commit

Commit the TAS profile

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tas_profile_commit(
    int unit,
    bcm_gport_t port,
    bcm_cosq_tas_profile_id_t pid)
```

PARAMETERS

unit	BCM device number
gport	(IN) gport id
pid	(IN) profile id

DESCRIPTION

Commit the profile to indicate the profile is ready to write to hardware.

Calendar entry auto adjustment will be proceed in bcm_cosq_tas_profile_commit when the spn_TAS_CALENDAR_AUTO_ADJUST_FOR_TXOVERRUN or spn_TAS_CALENDAR_AUTO_ADJUST_FOR_HOLDADVANCE is specified.

When Tx overrun (the transmission gate associated with a queue has closed and a frame that originated from the queue is still being transmitted) happens, TAS functionality will be impacted in the next hop. Delay parameters inside chip are required to be considered when programming the calendar entries to avoid TX overrun after gate closure. The sum of all the delays caused by different pipes is called Gate Close Response Time or Tgcr. The value specified in spn_TAS_CALENDAR_AUTO_ADJUST_REF_MAXSDU would be referenced to compose the device delay parameters Tgcr. NOTE: the value

of spn_TAS_CALENDAR_AUTO_ADJUST_REF_MAXSDU is only provide the information(queueMaxSDU)to calculte the delay parameters, SDK won't program any MTU constrain according to this soc property. (Assume frames exceeding queueMaxSDU will be filtered out)

When preemption(IEEE 802.1Qbu) is active and spn_TAS_CALENDAR_AUTO_ADJUST_FOR_HOLDADVANCE is specified, the entry adjustment would consider the holdAdvance time and the proper holdRequest configuration would be reflected to hold the preemptable traffic.

RETURNS

BCM_E_NONE

BCM_E_XXX

---

bcm_cosq_tas_profile_create

Create the TAS profile

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tas_profile_create(
    int unit,
    bcm_gport_t gport,
    bcm_cosq_tas_profile_t *profile,
    bcm_cosq_tas_profile_id_t *pid)
```

PARAMETERS

unit	BCM device number
gport	(IN) gport id
profile	(IN) Pointer to profile structure which specifies entries in calendar table, ptp time information for PTP mode
pid	(OUT) profile id

DESCRIPTION

Create a profile by specifying the profile settings including calendar table, ptp time information for PTP mode. A profile id will be assigned along with the settings.

RETURNS

BCM_E_NONE

BCM_E_XXX

---

bcm_cosq_tas_profile_destroy

Destroy the TAS profile

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tas_profile_destroy(
    int unit,
    bcm_gport_t gport,
    bcm_cosq_tas_profile_id_t pid)
```

PARAMETERS

unit	BCM device number
gport	(IN) gport id
pid(IN)	profile id

DESCRIPTION

Destroy the profile and associated resources.

RETURNS

BCM_E_NONE

BCM_E_XXX

bcm_cosq_tas_profile_get bcm_cosq_tas_profile_set

Get/set TAS profile information

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tas_profile_get(
    int unit,
    bcm_gport_t gport,
    bcm_cosq_tas_profile_id_t pid,
    bcm_cosq_tas_profile_t *profile)

int bcm_cosq_tas_profile_set(
    int unit,
    bcm_gport_t gport,
    bcm_cosq_tas_profile_id_t pid,
    bcm_cosq_tas_profile_t *profile)
```

PARAMETERS

unit	BCM device number
gport	(IN) gport id
pid	(IN) profile id
Profile	(IN)(for "_get") profile data
Profile	(OUT)(for "_set") profile data

DESCRIPTION

Get/set(modify) the profile information by specifying the profile id Note: for the set API, the profile could not be modified after invoking commit API.

RETURNS

BCM_E_NONE

BCM_E_XXX

bcm_cosq_tas_profile_status_get

Get the TAS profile status

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tas_profile_status_get(
    int unit,
    bcm_gport_t port,
    bcm_cosq_tas_profile_id_t pid,
    bcm_cosq_tas_profile_status_t *status)
```

PARAMETERS

unit	BCM device number
unit	(IN) unit number
gport	(IN) gport id
pid	(IN) profile id
status	(OUT) the profile status defined in cm_cosq_tas_profile_status_t

DESCRIPTION

Get the profile status like profile state, config change time and entries in the hardware calendar table.

RETURNS

BCM_E_NONE

BCM_E_XXX

bcm_cosq_tas_profile_status_t_init

Initialize the bcm_cosq_tas_profile_status_t structure

SYNOPSIS

```
#include <bcm/cosq.h>
void bcm_cosq_tas_profile_status_t_init(
    bcm_cosq_tas_profile_status_t *profile_status)
```

PARAMETERS

profile(IN)	Pointer to the bcm_cosq_tas_profile_status_t structure

DESCRIPTION

Initialize the bcm_cosq_tas_profile_status_t structure. Clear the content to zero.

bcm_cosq_tas_profile_t_init

Initialize the bcm_cosq_tas_profile_t structure

SYNOPSIS

```
#include <bcm/cosq.h>
void bcm_cosq_tas_profile_t_init(
    bcm_cosq_tas_profile_t  *profile)
```

PARAMETERS

profile(IN)	Pointer to the bcm_cosq_tas_profile_t structure

DESCRIPTION

Initialize the bcm_cosq_tas_profile_t structure. Clear the content to zero.

---

bcm_cosq_tas_profile_traverse

Traverse the TAS profile

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tas_profile_traverse(
    int unit,
    bcm_gport_t gport,
    bcm_cosq_tas_profile_traverse_cb cb,
    void *user_data);
```

PARAMETERS

unit	BCM device number
gport	(IN) gport id
cb	(IN) A pointer to callback function to call for each profile in the specified gport
user_data	(IN) Pointer to user data to supply in the callback

DESCRIPTION

Traverse the set of profiles associated with the specified port.

RETURNS

BCM_E_NONE

BCM_E_XXX

bcm_cosq_tas_status_get

Get the TAS status

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_tas_status_get(
    int unit,
    bcm_gport_t gport,
    bcm_cosq_tas_status_t type,
    int *arg)
```

PARAMETERS

unit	BCM device number
gport	(IN) gport id
type	(IN) tas status type defined in bcm_cosq_tas_status_t.
arg	(OUT) tas status.

DESCRIPTION

Define various status type when TAS is working, user can get the status by specifying the type which defined in bcm_cosq_tas_status_t. Each status type would be per port or per port per queue basis according to each chip capabilities. For the status type which is per port basis, the parameter cosq value should assign with -1.

TAS status type

```
typedef enum bcm_cosq_tas_status_e {
    bcmCosqTASStatusGateQueueState = 0, /*  The current gate state of each
                                           traffic class queue. The value
                                           represents all queues' gate state
                                           bitmap, each bit indicates to each
                                           queue. */
    bcmCosqTASStatusGateStateSet = 1,   /* Reflect which control gate source is
                                           selected. The value = 0 indicates the
                                           gate state is from INIT GATE STATE
                                           settings, value = 1 indicates the
                                           gate state is from active control
                                           list table, value = 2 indicates the
                                           gate state is from ERR GATE STATE
                                           settings */
    bcmCosqTASStatusTickGranularity = 2, /*  The granularity of the calendar
                                           table time clock in ns. */
    bcmCosqTASStatusMaxCalendarEntries = 3, /* The max entries could be support in
                                           calendar table. */
    bcmCosqTASSatusCount = 4            /* This should be the last one. */
} bcm_cosq_tas_status_t;
```

RETURNS

BCM_E_NONE

BCM_E_XXX

bcm_cosq_pfc_config_get bcm_cosq_pfc_config_set

Configure Priority-Based Flow Control (PFC).

SYNOPSIS

```
#include <bcm/cosq.h>
int bcm_cosq_pfc_config_set(
    int unit,
    bcm_gport_t gport,
    bcm_cos_queue_t cosq,
    uint32 flags,
    bcm_cosq_pfc_config_t *config);
```

```
int bcm_cosq_pfc_config_get(
    int unit,
    bcm_gport_t gport,
    bcm_cos_queue_t cosq,
    uint32 flags,
    bcm_cosq_pfc_config_t *config);
```

PARAMETERS

unit	BCM device number
gport	Source port for Source port/Priority based flow control scheme, pass in BCM_GPORT_INVALID for Priority only based flow control scheme.
cosq	Packet priority
flags	Specifies configuration. Valid flags: BCM_COSQ_GPORT_PFC_CONFIG_WITH_ID
config	PFC configurations. see table

DESCRIPTION

PFC configuration is listed in the following tables.

Table 6.7: BCM CoSQ PFC Configuration

Parameter	Description
xoff_threshold	Assert backpressure when number of buffers used is at or above this threshold
xon_threshold	Remove backpressure when number of buffers used drops below this threshold
xoff_fadt_threshold	Assert backpressure when number of resources used is at or above this FADT threshold
xon_fadt_offset	Remove backpressure when number of resources used drops below FADT threshold defined by xoff_fadt_threshold - fadt_offset
xoff_threshold_bd	Assert backpressure when number of buffer descriptors used is at or above this threshold
xon_threshold_bd	Remove backpressure when number of buffer descriptors used drops below this threshold
drop_threshold	Drop packets when number of buffers used hit this threshold

reserved_buffers	Number of buffers reserved for the specified priority
lossless	lossless PFC enable for the specified priority

Configure the priority-based flow control (PFC) features. When gport is set to BCM_GPORT_INVALID, the API configs the thresholds for the priority specified by the cosq parameter. For (source port, priority) based flow control scheme, the gport specifies the source port. It is possible that devices support both per priority threshold and (source port, priority) threshold at the same time.

Typical settings for the thresholds are drop_threshold >= xoff_threshold >= xon_threshold, When the buffers used for a priority hit or go above the xoff_threshold, backpressure is asserted, if the buffers used are at drop_threshld, all incoming packets will be dropped. when the buffers used for the priority drops below the xon_threshold, backpressure is removed. drop_threshold could be disabled when set to 0.

For a particular device, it is possible only some config parameters applies, unused configs will be ignored by SDK.

```
#define BCM_COSQ_PFC_BYTES /* threshold in bytes */
#define BCM_COSQ_PFC_OCB /* threshold for OCB (SRAM) resources */
#define BCM_COSQ_PFC_PACKET_DESC /* threshold in packet descriptors */
#define BCM_COSQ_PFC_BUFFERS /* threshold in buffers */
```

Section 6.2: Field Processor

Table 6.8: New Field Qualifiers

bcmFieldQualifyPreemptablePacket	Preemptable Packet
bcmFieldQualifyVxlanClassValid	VXLAN Class is valid or not
bcmFieldQualifyVxlanPacket	Indicates whether it is a VXLAN packet
bcmFieldQualifyVxlanVnidVlanTranslateHit	VXLAN Network Identifier (VN_ID) lookup hit through vlan translation
bcmFieldQualifyVxlanPayloadVlanFormat	VLAN tag format in VXLAN payload
bcmFieldQualifyTunnelPayloadDstMac	Destination MAC address in Tunnel payload
bcmFieldQualifyTunnelPayloadSrcMac	Source MAC address in Tunnel payload
bcmFieldQualifyTunnelPayloadEtherType	Ethernet type in Tunnel payload
bcmFieldQualifyTunnelPayloadOuterVlan	Outer VLAN tag in Tunnel payload
bcmFieldQualifyTunnelPayloadOuterVlanId	Outer VLAN id in Tunnel payload
bcmFieldQualifyTunnelPayloadOuterVlanPri	Outer VLAN priority in Tunnel payload
bcmFieldQualifyTunnelPayloadOuterVlanCfi	Outer VLAN CFI in Tunnel payload
bcmFieldQualifyTunnelPayloadSip	IPv4 source address in Tunnel payload
bcmFieldQualifyTunnelPayloadDip	IPv4 destination address in Tunnel payload
bcmFieldQualifyTunnelPayloadSip6	IPv6 source address in Tunnel payload
bcmFieldQualifyTunnelPayloadDip6	IPv6 destination address in Tunnel payload
bcmFieldQualifyTunnelPayloadIpProtocol	IP Protocol value in Tunnel payload
bcmFieldQualifyTunnelPayloadL4DstPort	UDP/TCP destination port number in Tunnel payload
bcmFieldQualifyTunnelPayloadL4SrcPort	UDP/TCP source port number in Tunnel payload
bcmFieldQualifyRoceBthOpcode	To qualify on OP code on RoCEv1 / RoCEv2 Base Transport Header(BTH)
bcmFieldQualifyRoceBthPartitionKey	To qualify on partition key on RoCEv1 / RoCEv2 Base Transport Header (BTH)
bcmFieldQualifyRoceBthDstQueuePair	To qualify on destination queue pair on RoCEv1 / RoCEv2 Base Transport Header(BTH)
bcmFieldQualifyRoceBthFlags	To qualify on flags on RoCEv1 / RoCEv2 Base

	Transport Header(BTH)
bcmFieldQualifyRoceVer1Pkt	Indicate whether it is a RoCEv1 packet
bcmFieldQualifyRoceVer2Pkt	Indicate whether it is a RoCEv2 packet
bcmFieldQualifySrcPortSRType	SR (Seamless Redundancy) port type (bcmFieldPortSRTypeXXX) assigned to the source port
bcmFieldQualifyDstPortSRType	SR (Seamless Redundancy) port type (bcmFieldPortSRTypeXXX) assigned to the destination port
bcmFieldQualifySrcPortSRRoleInterlink	Indicates whether the SR role for the source is an interlink
bcmFieldQualifyDstPortSRRoleInterlink	Indicates whether the SR role for the destination is an interlink
bcmFieldQualifySrcPortSRMode	The SR port mode (bcmFieldSRPortModeXXX) assigned to the source port
bcmFieldQualifyDstPortSRMode	The SR port mode (bcmFieldSRPortModeXXX) assigned to the destination port
bcmFieldQualifySrcPortSRNetId	The SR NET ID (0~7) assigned to the source port
bcmFieldQualifyDstPortSRNetId	The SR NET ID (0~7) assigned to the destination port
bcmFieldQualifySrcPortSRLanId	The SR LAN ID (0 for LAN A and 1 for LAN B) assigned to the source port
bcmFieldQualifyDstPortSRLanId	The SR LAN ID (0 for LAN A and 1 for LAN B) assigned to the destination port
bcmFieldQualifySRTagType	The SR (Seamless Redundancy) tag type (bcmFieldSRTagTypeXXX) identified in this packet
bcmFieldQualifySRLanId	The SR LAN ID (0 for LAN A and 1 for LAN B) in the SR tag. Valid only if the packet contains an SR tag.
bcmFieldQualifySRNetId	The SR NET ID (0~7) in the SR tag. Valid only if the packet contains an SR tag.
bcmFieldQualifyVlanSREnable	Indicates whether SR (Seamless Redundancy) is enabled in this VLAN
bcmFieldQualifyVlanSRLanId	Indicates the SR LAN ID (0 for LAN A and 1 for LAN B) for this VLAN. Valid only if SR is enabled for this VLAN.
bcmFieldQualifySRFlowId	SR flow ID for the SR flow that the packet belongs to

bcmFieldQualifyL2DestSRNodeType	SR node type (bcmFieldSRNodeTypeXXX) for the destination address
bcmFieldQualifySRNetIdMatched	Indicates whether the SR packet's NET ID matches the configured NET ID on the destination port
bcmFieldQualifySRSrcNodeIsSan	Indicates whether the source address belongs to an SR SAN node
bcmFieldQualifySRSupervisionType	The SR supervision type (bcmFieldSRSupervisionTypeXXX) if it's an SR supervision packet
bcmFieldQualifySRError	Indicates SR errors for the packet
bcmFieldQualifyL2SrcMulticastHit	Indicates L2 source address found on the correct multicast group.
bcmFieldQualifyL2DstMulticastHit	Indicates L2 destination address found and result is a multicast group.
bcmFieldQualifySRDuplicate	SR (Seamless Redundancy): Indicates whether this packet is a duplicate SR packet
bcmFieldQualifyTsnFlowId	TSN (Time-Sensitive Networking) flow ID for the TSN flow that the packet belongs to
bcmFieldQualifyExternalValue6	External lookup 6 value
bcmFieldQualifyExternalValue7	External lookup 7 value
bcmFieldQualifyExternalHit6	External lookup 6 hit
bcmFieldQualifyExternalHit7	External lookup 7 hit

Table 6.9: New members in Field Application Type

Type	Purpose
bcmFieldAppTypeIp6UcastPublic	IPv6 Unicast Public Routing.
bcmFieldAppTypeIp6UcastRpfPublic	IPv6 Unicast Public Routing (coupled to the RPF lookup).

Section 6.2: Field Processor

bcm_field_qualify_PreemptablePacket bcm_field_qualify_PreemptablePacket_get

Set/Get match criteria for bcmFieldQualifyPreemptablePacket Preemptable Packet

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_PreemptablePacket(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_PreemptablePacket_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

RETURNS

int

bcm_field_qualify_VxlanClassValid bcm_field_qualify_VxlanClassValid_get

Set/Get match criteria for bcmFieldQualifyVxlanClassValid VXLAN Class is valid or not

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_VxlanClassValid(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_VxlanClassValid_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

RETURNS

int

bcm_field_qualify_VxlanPacket bcm_field_qualify_VxlanPacket_get

Get/Set/Get match criteria for bcmFieldQualifyVxlanPacket Indicates whether it is a VXLAN packet

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_VxlanPacket(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_VxlanPacket_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

RETURNS

int

---

bcmFieldQualifyVxlanVnidVlanTranslateHit bcmFieldQualifyVxlanVnidVlanTranslateHit_get

Set/Get match criteria for bcmFieldQualifyVxlanVnidVlanTranslateHit VXLAN Network Identifier (VN_ID) lookup hit through vlan translation

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_VxlanVnidVlanTranslateHit(
    int unit,
```

```
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_VxlanVnidVlanTranslateHit_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

RETURNS

int

---

bcm_field_qualify_VxlanPayloadVlanFormat bcm_field_qualify_VxlanPayloadVlanFormat_get

Set/Get match criteria for bcmFieldQualifyVxlanPayloadVlanFormat VLAN tag format in VXLAN payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_VxlanPayloadVlanFormat(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_VxlanPayloadVlanFormat_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadDstMac bcm_field_qualify_TunnelPayloadDstMac_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadDstMac Destination MAC address in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadDstMac(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadDstMac_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadSrcMac bcm_field_qualify_TunnelPayloadSrcMac_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadSrcMac Source MAC address in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadSrcMac(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadSrcMac_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadEtherType bcm_field_qualify_TunnelPayloadEtherType_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadEtherType Ethernet type in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadEtherType(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadEtherType_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadOuterVlan bcm_field_qualify_TunnelPayloadOuterVlan_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadOuterVlan Outer VLAN tag in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadOuterVlan(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadOuterVlan_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

RETURNS

int

bcm_field_qualify_TunnelPayloadOuterVlanId bcm_field_qualify_TunnelPayloadOuterVlanId_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadOuterVlanId Outer VLAN id in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadOuterVlanId(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadOuterVlanId_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadOuterVlanPri bcm_field_qualify_TunnelPayloadOuterVlanPri_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadOuterVlanPri Outer VLAN priority in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadOuterVlanPri(
    int unit,
```

```
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadOuterVlanPri_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadOuterVlanCfi bcm_field_qualify_TunnelPayloadOuterVlanCfi_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadOuterVlanCfi Outer VLAN CFI in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadOuterVlanCfi(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadOuterVlanCfi_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadSip bcm_field_qualify_TunnelPayloadSip_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadSip IPv4 source address in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadSip(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadSip_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadDip bcm_field_qualify_TunnelPayloadDip_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadDip IPv4 destination address in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadDip(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadDip_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadSip6 bcm_field_qualify_TunnelPayloadSip6_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadSip6 IPv6 source address in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadSip6(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadSip6_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadDip6 bcm_field_qualify_TunnelPayloadDip6_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadDip6 IPv6 destination address in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadDip6(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadDip6_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

RETURNS

int

bcm_field_qualify_TunnelPayloadIpProtocol bcm_field_qualify_TunnelPayloadIpProtocol_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadIpProtocol IP Protocol value in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadIpProtocol(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadIpProtocol_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadL4DstPort bcm_field_qualify_TunnelPayloadL4DstPort_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadL4DstPort UDP/TCP destination port number in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadL4DstPort(
    int unit,
```

```
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadL4DstPort_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

RETURNS

int

---

bcm_field_qualify_TunnelPayloadL4SrcPort bcm_field_qualify_TunnelPayloadL4SrcPort_get

Set/Get match criteria for bcmFieldQualifyTunnelPayloadL4SrcPort UDP/TCP source port number in Tunnel payload

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TunnelPayloadL4SrcPort(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TunnelPayloadL4SrcPort_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

RETURNS

int

---

bcm_field_qualify_RoceBthOpcode bcm_field_qualify_RoceBthOpcode_get

Set/Get match criteria for bcmFieldQualifyRoceBthOpcode To qualify on OP code on RoCEv1 / RoCEv2 Base Transport Header(BTH)

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_RoceBthOpcode(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_RoceBthOpcode_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

RETURNS

int

---

bcm_field_qualify_RoceBthPartitionKey bcm_field_qualify_RoceBthPartitionKey_get

Set/Get match criteria for bcmFieldQualifyRoceBthPartitionKey To qualify on partition key on RoCEv1 / RoCEv2 Base Transport Header (BTH)

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_RoceBthPartitionKey(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_RoceBthPartitionKey_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

RETURNS

int

---

bcm_field_qualify_RoceBthDstQueuePair bcm_field_qualify_RoceBthDstQueuePair_get

Set/Get match criteria for bcmFieldQualifyRoceBthDstQueuePair To qualify on destination queue pair on RoCEv1 / RoCEv2 Base Transport Header(BTH)

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_RoceBthDstQueuePair(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_RoceBthDstQueuePair_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

RETURNS

int

---

bcm_field_qualify_RoceBthFlags bcm_field_qualify_RoceBthFlags_get

Set/Get match criteria for bcmFieldQualifyRoceBthFlags To qualify on flags on RoCEv1 / RoCEv2 Base Transport Header(BTH)

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_RoceBthFlags(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_RoceBthFlags_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

RETURNS

int

bcm_field_qualify_RoceVer1Pkt bcm_field_qualify_RoceVer1Pkt_get

Set/Get match criteria for bcmFieldQualifyRoceVer1Pkt Indicate whether it is a RoCEv1 packet

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_RoceVer1Pkt(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_RoceVer1Pkt_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

RETURNS

int

---

bcm_field_qualify_RoceVer2Pkt bcm_field_qualify_RoceVer2Pkt_get

Set/Get match criteria for bcmFieldQualifyRoceVer2Pkt Indicate whether it is a RoCEv2 packet

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_RoceVer2Pkt(
    int unit,
```

```
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_RoceVer2Pkt_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

RETURNS

int

---

bcm_field_qualify_SrcPortSRType bcm_field_qualify_SrcPortSRType_get

Set/Get match criteria for bcmFieldQualifySrcPortSRType SR (Seamless Redundancy) port type (bcmFieldPortSRTypeXXX) assigned to the source port

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SrcPortSRType(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SrcPortSRType_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

RETURNS

int

---

bcm_field_qualify_DstPortSRType bcm_field_qualify_DstPortSRType_get

Set/Get match criteria for bcmFieldQualifyDstPortSRType SR (Seamless Redundancy) port type (bcmFieldPortSRTypeXXX) assigned to the destination port

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_DstPortSRType(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_DstPortSRType_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

RETURNS

int

---

bcm_field_qualify_SrcPortSRRoleInterlink bcm_field_qualify_SrcPortSRRoleInterlink_get

Set/Get match criteria for bcmFieldQualifySrcPortSRRoleInterlink Indicates whether the SR role for the source is an interlink

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SrcPortSRRoleInterlink(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SrcPortSRRoleInterlink_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

RETURNS

int

---

bcm_field_qualify_DstPortSRRoleInterlink bcm_field_qualify_DstPortSRRoleInterlink_get

Set/Get match criteria for bcmFieldQualifyDstPortSRRoleInterlink Indicates whether the SR role for the destination is an interlink

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_DstPortSRRoleInterlink(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_DstPortSRRoleInterlink_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

RETURNS

int

bcm_field_qualify_SrcPortSRMode bcm_field_qualify_SrcPortSRMode_get

Set/Get match criteria for bcmFieldQualifySrcPortSRMode The SR port mode (bcmFieldSRPortModeXXX) assigned to the source port

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SrcPortSRMode(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SrcPortSRMode_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

RETURNS

int

bcm_field_qualify_DstPortSRMode bcm_field_qualify_DstPortSRMode_get

Set/Get match criteria for bcmFieldQualifyDstPortSRMode The SR port mode (bcmFieldSRPortModeXXX) assigned to the destination port

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_DstPortSRMode(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_DstPortSRMode_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

RETURNS

int

---

bcm_field_qualify_SrcPortSRNetId bcm_field_qualify_SrcPortSRNetId_get

Set/Get match criteria for bcmFieldQualifySrcPortSRNetId The SR NET ID (0~7) assigned to the source port

SYNOPSIS

```
#include <bcm/field.h>
```

```
int
bcm_field_qualify_SrcPortSRNetId(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SrcPortSRNetId_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

RETURNS

int

---

bcm_field_qualify_DstPortSRNetId bcm_field_qualify_DstPortSRNetId_get

Set/Get match criteria for bcmFieldQualifyDstPortSRNetId The SR NET ID (0~7) assigned to the destination port

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_DstPortSRNetId(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
```

```
bcm_field_qualify_DstPortSRNetId_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

RETURNS

int

---

bcm_field_qualify_SrcPortSRLanId bcm_field_qualify_SrcPortSRLanId_get

Set/Get match criteria for bcmFieldQualifySrcPortSRLanId The SR LAN ID (0 for LAN A and 1 for LAN B) assigned to the source port

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SrcPortSRLanId(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SrcPortSRLanId_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

RETURNS

int

---

bcm_field_qualify_DstPortSRLanId bcm_field_qualify_DstPortSRLanId_get

Set/Get match criteria for bcmFieldQualifyDstPortSRLanId The SR LAN ID (0 for LAN A and 1 for LAN B) assigned to the destination port

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_DstPortSRLanId(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_DstPortSRLanId_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

RETURNS

int

---

bcm_field_qualify_SRTagType bcm_field_qualify_SRTagType_get

Set/Get match criteria for bcmFieldQualifySRTagType The SR (Seamless Redundancy) tag type (bcmFieldSRTagTypeXXX) identified in this packet

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SRTagType(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SRTagType_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

RETURNS

int

bcm_field_qualify_SRLanId bcm_field_qualify_SRLanId_get

Set/Get match criteria for bcmFieldQualifySRLanId The SR LAN ID (0 for LAN A and 1 for LAN B) in the SR tag. Valid only if the packet contains an SR tag.

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SRLanId(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SRLanId_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

RETURNS

int

bcm_field_qualify_SRNetId bcm_field_qualify_SRNetId_get

Set/Get match criteria for bcmFieldQualifySRNetId The SR NET ID (0~7) in the SR tag. Valid only if the packet contains an SR tag.

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SRNetId(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SRNetId_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

RETURNS

int

---

bcm_field_qualify_VlanSREnable bcm_field_qualify_VlanSREnable_get

Set/Get match criteria for bcmFieldQualifyVlanSREnable Indicates whether SR (Seamless Redundancy) is enabled in this VLAN

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_VlanSREnable(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
```

```
    uint32 *mask);

int
bcm_field_qualify_VlanSREnable_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

RETURNS

int

bcm_field_qualify_VlanSRLanId bcm_field_qualify_VlanSRLanId_get

Set/Get match criteria for bcmFieldQualifyVlanSRLanId Indicates the SR LAN ID (0 for LAN A and 1 for LAN B) for this VLAN. Valid only if SR is enabled for this VLAN.

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_VlanSRLanId(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_VlanSRLanId_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

RETURNS

int

bcm_field_qualify_SRFlowId bcm_field_qualify_SRFlowId_get

Set/Get match criteria for bcmFieldQualifySRFlowId SR flow ID for the SR flow that the packet belongs to

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SRFlowId(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SRFlowId_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

RETURNS

int

---

bcm_field_qualify_L2DestSRNodeType bcm_field_qualify_L2DestSRNodeType_get

Set/Get match criteria for bcmFieldQualifyL2DestSRNodeType SR node type (bcmFieldSRNodeTypeXXX) for the destination address

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_L2DestSRNodeType(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_L2DestSRNodeType_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

RETURNS

int

---

bcm_field_qualify_SRNetIdMatched bcm_field_qualify_SRNetIdMatched_get

Set/Get match criteria for bcmFieldQualifySRNetIdMatched Indicates whether the SR packet's NET ID matches the configured NET ID on the destination port

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SRNetIdMatched(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SRNetIdMatched_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

RETURNS

int

---

bcm_field_qualify_SRSrcNodeIsSan bcm_field_qualify_SRSrcNodeIsSan_get

Set/Get match criteria for bcmFieldQualifySRSrcNodeIsSan Indicates whether the source address belongs to an SR SAN node

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SRSrcNodeIsSan(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SRSrcNodeIsSan_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

RETURNS

int

bcm_field_qualify_SRSupervisionType bcm_field_qualify_SRSupervisionType_get

Set/Get match criteria for bcmFieldQualifySRSupervisionType The SR supervision type (bcmFieldSRSupervisionTypeXXX) if it's an SR supervision packet

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SRSupervisionType(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SRSupervisionType_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

RETURNS

int

---

bcm_field_qualify_SRError bcm_field_qualify_SRError_get

Set/Get match criteria for bcmFieldQualifySRError Indicates SR errors for the packet

SYNOPSIS

```
#include <bcm/field.h>

int
```

```
bcm_field_qualify_SRError(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SRError_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

RETURNS

int

---

bcm_field_qualify_L2SrcMulticastHit bcm_field_qualify_L2SrcMulticastHit_get

Set/Get match criteria for bcmFieldQualifyL2SrcMulticastHit Indicates L2 source address found on the correct multicast group.

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_L2SrcMulticastHit(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_L2SrcMulticastHit_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

RETURNS

int

---

bcm_field_qualify_L2DstMulticastHit bcm_field_qualify_L2DstMulticastHit_get

Set/Get match criteria for bcmFieldQualifyL2DstMulticastHit Indicates L2 destination address found and result is a multicast group.

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_L2DstMulticastHit(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_L2DstMulticastHit_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

RETURNS

int

---

bcm_field_qualify_SRDuplicate bcm_field_qualify_SRDuplicate_get

Set/Get match criteria for bcmFieldQualifySRDuplicate SR (Seamless Redundancy): Indicates whether this packet is a duplicate SR packet

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_SRDuplicate(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_SRDuplicate_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

RETURNS

int

---

bcm_field_qualify_TsnFlowId
bcm_field_qualify_TsnFlowId_get

Set/Get match criteria for bcmFieldQualifyTsnFlowId TSN (Time-Sensitive Networking) flow ID for the TSN flow that the packet belongs to

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_TsnFlowId(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_TsnFlowId_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

RETURNS

int

bcm_field_qualify_ExternalValue6
bcm_field_qualify_ExternalValue6_get

Set/Get match criteria for bcmFieldQualifyExternalValue6 External lookup 6 value

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_ExternalValue6(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_ExternalValue6_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

RETURNS

int

bcm_field_qualify_ExternalValue7 bcm_field_qualify_ExternalValue7_get

Set/Get match criteria for bcmFieldQualifyExternalValue7 External lookup 7 value

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_ExternalValue7(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_ExternalValue7_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

RETURNS

int

bcm_field_qualify_ExternalHit6 bcm_field_qualify_ExternalHit6_get

Set/Get match criteria for bcmFieldQualifyExternalHit6 External lookup 6 hit

SYNOPSIS

```
#include <bcm/field.h>

int
bcm_field_qualify_ExternalHit6(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_ExternalHit6_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

RETURNS

int

---

bcm_field_qualify_ExternalHit7 bcm_field_qualify_ExternalHit7_get

Set/Get match criteria for bcmFieldQualifyExternalHit7 External lookup 7 hit

SYNOPSIS

```
#include <bcm/field.h>
```

```
int
bcm_field_qualify_ExternalHit7(
    int unit,
    bcm_field_entry_t entry,
    uint32 *data,
    uint32 *mask);

int
bcm_field_qualify_ExternalHit7_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

RETURNS

int

Section 6.3: Kernel Network (KNET) Configuration

Table 6.10: BCM KNET Network Interface Flags

Name	Purpose	
BCM_KNET_NETIF_F_KEEP_RX_TAG	Keep tag of packets which will be sent to this interface	New

Section 6.4: Layer 2 Address Management

New members were added to L2 address table entries.

```
typedef struct bcm_l2_addr_s {
     uint32              flags;          /* BCM_L2_XXX flags */
     uint32              station_flags;  /* BCM_L2_STATION_xxx flags */
     bcm_mac_t           mac;            /* 802.3 MAC address */
     bcm_vlan_t          vid;            /* VLAN or VPN identifier */
     int                 port;           /* Zero-based port number */
     int                 modid;          /* XGS: modid */
     bcm_trunk_t         tgid;           /* Trunk group ID */
     bcm_cos_t           cos_dst;        /* CoS based on dst addr */
     bcm_cos_t           cos_src;        /* CoS based on src addr */
     bcm_multicast_t     l2mc_group;     /* XGS: index in L2MC table */
     bcm_pbmp_t          block_bitmap;   /* XGS: blocked egress bitmap */
     int                 auth;           /* Used if auth enabled on port */
     int                 group;          /* Group number for FP */
     bcm_fabric_distribution_t distribution_class; /* Fabric Distribution Class.
*/
     int encap_id;                       /* out logical interface*/
     int age_state;                      /* Age state of the entry */
+    bcm_pbmp_t sa_source_filter_pbmp;   /* Source port filter bitmap for this SA
*/
+    bcm_tsn_flowset_t tsn_flowset;      /* Time-Sensitive Networking: associated
+                                           flow set */
+    bcm_tsn_sr_flowset_t sr_flowset;    /* Seamless Redundancy: associated flow
+                                           set */
} bcm_l2_addr_t;
```

Section 6.5: Layer 3 Management

New member added to bcm_l3_ingress_t

```
typedef struct bcm_l3_ingress_s {
     uint32 flags;               /* Interface flags. */
     bcm_vrf_t vrf;              /* Virtual router ID. */
     bcm_l3_ingress_urpf_mode_t urpf_mode; /* URPF mode setting for L3-IIF */
     int intf_class;             /* Classification class ID. */
     bcm_vlan_t ipmc_intf_id;    /* IPMC L2 distribution Vlan. */
     int qos_map_id;             /* QoS DSCP priority map. */
     int ip4_options_profile_id; /* Profile ID for handling IP options */
     int nat_realm_id;           /* Realm id of the interface for NAT */
     int tunnel_term_ecn_map_id; /* Tunnel termination ecn map id */
+    uint32 intf_class_route_disable;    /* routing enablers bit map in rif
+                                           profile */

 } bcm_l3_ingress_t;
```

Section 6.6: Link Bonding (LB)

Table 6.11: LB Miscellaneous defines

Name	Description	
BCM_LB_FLAG_FLUSH_GLOBAL	Flush operation for all Link Bonding Group.	New
BCM_LB_FLAG_MODEM_SHAPER_SEGMENT	If this flag is set, then configure modem segment shaper. otherwise, configure bit shaper.	New

Table 6.12: LB control type

control type	Description	
bcmLbControlNofGuaranteedBuffers	Guaranteed buffers the link bonding group can acquire	New
bcmLbControlNofTotalSharedBuffers	Total shared buffers	New

bcm_lb_flush_configure_set bcm_lb_flush_configure_get

Set/get flush configuration at LBG or global level.

SYNOPSIS

```
#include <bcm/lb.h>
int bcm_lb_flush_configure_set(
    int unit,
    bcm_lbg_t lbg_id,
    uint32 flags,
    bcm_lb_flush_type_t type,
    int arg);
int bcm_lb_flush_configure_get(
    int unit,
    bcm_lbg_t lbg_id,
    uint32 flags,
    bcm_lb_flush_type_t type,
    int *arg);
```

PARAMETERS

unit	(IN) BCM device number
lbg_id	(IN) LBG ID
flags	(IN) BCM_LB_FLAG_FLUSH_GLOBAL : global configuration
type	(IN) type of operation (see (unknown XREF BCM_LB_FLUSH_OPERATION_TYPE_table))
arg	(IN) (for "_set")Argument whose meaning is dependent on type
arg	(OUT) (for "_get")Argument whose meaning is dependent on type

DESCRIPTION

Features that can be controlled on LBG/global basis.

Link bonding flush operation type

```
typedef enum bcm_lb_flush_type_e {
    bcmLbFlushTypeForceFlush = 0,       /* For debug only. Should not be used in
                                           normal operation */
    bcmLbFlushTypeFlushOnGroupOverflow = 1,
    bcmLbFlushTypeFlushOnFifoOverflow = 2,
    bcmLbFlushTypeFlushOnSharedOverflow = 3,
    bcmLbFlushTypeCount = 4             /* Always Last. Not a usable value. */
} bcm_lb_flush_type_t;
```

RETURNS

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set/get configuration with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

Section 6.7: MPLS Management

Table 6.13: MPLS Port Flags(2)

Name	Purpose	
BCM_MPLS_PORT2_KEEP_TTL	Retain incoming TTL by disabling decrement for MPLS flows.	New

Table 6.14: MPLS Tunnel Switch Flags

Name	Purpose	
BCM_MPLS_SWITCH_KEEP_TTL	Retain incoming TTL by disabling decrement for MPLS flows.	New

Section 6.8: Operations, Administration, And Maintenance

Table 6.15: OAM Group Structure Flag Definitions

Flag	Description	
BCM_OAM_GROUP_RX_NAME	This group has two MAIDs, one for packets specified in the 'name' field, and the MAID for packets received is specified in the 'rx_name' field	New

Additional member to OAM group info structure. This is equivalent to an 802.1ag Maintenance Association (MA):

```
typedef struct bcm_oam_group_info_s {
     uint32 flags;
     bcm_oam_group_t id;
     uint8 name[BCM_OAM_GROUP_NAME_LENGTH]; /* This is equivalent to the
Maintenance
                                               Association ID (MAID) in 802.1ag.
```

```
                                          Note that this is not a
                                          null-terminated string but an array
                                          of bytes.  To avoid memory
                                          corruption, do not use string copy to
                                          populate this field. */
+    uint8 rx_name[BCM_OAM_GROUP_NAME_LENGTH]; /* Additional MAID used for
received OAM
+                                           packets. It is used when the flag
+                                           BCM_OAM_GROUP_RX_NAME is set.  When
+                                           used, the received MAID values are
+                                           compared to this MAID instead of the
+                                           'name' field. */
     uint32 faults;                      /* Fault flags */
     uint32 persistent_faults;           /* Persistent fault flags */
     uint32 clear_persistent_faults;     /* Persistent fault flags to clear on a
                                            'get' */
   bcm_oam_group_fault_alarm_defect_priority_t lowest_alarm_priority; /* Generate
                                                                         fault alarm
for this
                                                                         maintenance
endpoint when defects of
                                                                         greater than
or equal to this
                                                                         priority
value are detected on this
                                                                         maintenance
endpoint */
    int group_name_index;              /* Pointer to first extended data entry */
 } bcm_oam_group_info_t;
```

Section 6.9: Rate Limiting

bcm_rate_packet_set bcm_rate_packet_get

Front ends to bcm_rate_packet_set/get functions. Uses a single data structure to write into rate control registers

SYNOPSIS

```
#include <bcm/rate.h>
int bcm_rate_packet_set(int unit, bcm_port_t port, bcm_rate_packet_t *pkt_rate);
int bcm_rate_packet_get(int unit, bcm_port_t port, bcm_rate_packet_t *pkt_rate);
```

PARAMETERS

unit	BCM device number
port	BCM port number
pkt_rate	(for "_set") Data structure containing info to be written to the rate control registers
pkt_rate	(for "_get", OUT) Data structure containing info from rate control registers

DESCRIPTION

Configure or retrieve different rate packet for each of the three packet types.

```
/* bcm_rate_packet_s */
typedef struct bcm_rate_packet_s {
  uint32       flags;
  int           pps;
  int           kbits_burst;
} bcm_rate_packet_t;
```

RETURNS

BCM_E_NONE	Success.
BCM_E_UNAVAIL	Not supported.

BCM_E_XXX	Other possible errors; for details, see (unknown XREF error_codes)

bcm_rate_packet_t_init

Initialize a bcm_rate_packet_t structure.

SYNOPSIS

```
#include <bcm/rate.h>
void bcm_rate_packet_t_init(bcm_rate_packet *pkt_rate);
```

PARAMETERS

pkt_rate	(OUT) Pointer to the bcm_rate_packet_t structure to initialize.

DESCRIPTION

Initializes a Rate Packet structure to default values. This function should be used to initialize any rate packet structure prior to filling it out and passing it to an API function. This ensures that subsequent API releases may add new structure members to the bcm_rate_packet_t structure, and bcm_rate_packet_t_init will initialize the new members to correct default values.

RETURNS

Nothing

Section 6.10: Switch Control

Extended flags for bcmSwitchHashSelectControlExtended switch control type to disable enhanced hash field selection control.

BCM_HASH_FIELD0_DISABLE_ROCE1 - Hash block A field selection disable for RoCEv1 packet
BCM_HASH_FIELD1_DISABLE_ROCE1 - Hash block B field selection disable for RoCEv1 packet
BCM_HASH_FIELD0_DISABLE_ROCE2_IP4 - Hash block A field selection disable for RoCEv2 IPv4 packet
BCM_HASH_FIELD1_DISABLE_ROCE2_IP4 - Hash block B field selection disable for RoCEv2 IPv4 packet
BCM_HASH_FIELD0_DISABLE_ROCE2_IP6 - Hash block A field selection disable for RoCEv2 IPv6 packet
BCM_HASH_FIELD1_DISABLE_ROCE2_IP6 - Hash block B field selection disable for RoCEv2 IPv6 packet

Enhanced field selection bit map for RoCE hash field selection.

BCM_HASH_ROCE_FIELD_DSTMOD - Destination port ID
BCM_HASH_ROCE_FIELD_DSTPORT - Destination port ID
BCM_HASH_ROCE_FIELD_SRCMOD - Source module ID
BCM_HASH_ROCE_FIELD_SRCPORT - Source port ID
BCM_HASH_ROCE_FIELD_BTH_DSTQP_COLLAPSED - RoCE Collapsed Destination Queue Pair in BTH (Base Transport Header)
BCM_HASH_ROCE_FIELD_BTH_PARTITION_KEY - RoCE Partition Key in BTH (Base Transport Header)
BCM_HASH_ROCE_FIELD_VLAN - VLAN ID
BCM_HASH_ROCE_FIELD_MACDA_LO - RoCEv1 MAC destination address lower 16 bits
BCM_HASH_ROCE_FIELD_MACDA_MI - RoCEv1 MAC destination address middle 16 bits
BCM_HASH_ROCE_FIELD_MACDA_HI - RoCEv1 MAC destination address upper 16 bits
BCM_HASH_ROCE_FIELD_MACSA_LO - RoCEv1 MAC source address lower 16 bits
BCM_HASH_ROCE_FIELD_MACSA_MI - RoCEv1 MAC source address middle 16 bits
BCM_HASH_ROCE_FIELD_MACSA_HI - RoCEv1 MAC source address upper16 bits

Hash block A field selection for RoCEv2 IPv4 packets.

BCM_HASH_ROCE_FIELD_DSTMOD - Destination port ID
BCM_HASH_ROCE_FIELD_DSTPORT - Destination port ID
BCM_HASH_ROCE_FIELD_SRCMOD - Source module ID
BCM_HASH_ROCE_FIELD_SRCPORT - Source port ID
BCM_HASH_ROCE_FIELD_BTH_DSTQP_COLLAPSED - RoCE Collapsed Destination Queue Pair in BTH (Base Transport Header)
BCM_HASH_ROCE_FIELD_BTH_PARTITION_KEY - RoCE Partition Key in BTH (Base Transport Header)
BCM_HASH_ROCE_FIELD_DSTL4 - RoCEv2 L4 destination port

BCM_HASH_ROCE_FIELD_SRCL4 - RoCEv2 L4 source port
BCM_HASH_ROCE_FIELD_VLAN - VLAN ID
BCM_HASH_ROCE_FIELD_IP4DST_LO - RoCEv2 IPv4 destination address lower 16 bits
BCM_HASH_ROCE_FIELD_IP4DST_HI - RoCEv2 IPv4 destination address upper 16 bits
BCM_HASH_ROCE_FIELD_IP4SRC_LO - RoCEv2 IPv4 source address lower 16 bits
BCM_HASH_ROCE_FIELD_IP4SRC_HI - RoCEv2 IPv4 source address upper 16 bits

New member was added to bcm_switch_tpid_type_t used in bcm_switch_tpid_add

```
typedef enum bcm_switch_tpid_type_e {
     bcmTpidTypeOuter,   /* Global Outer TPID */
     bcmTpidTypeInner,   /* Global Inner TPID */
     bcmTpidTypeVntag,   /* Global TPID for Vntag */
     bcmTpidTypeEtag,    /* Global TPID for Etag */
     bcmTpidTypeCapwapPayloadOuter,     /* Global Outer TPID of CAPWAP wireless
payload */
     bcmTpidTypeCapwapPayloadInner,     /* Global Inner TPID of CAPWAP wireless
payload */
     bcmTpidTypeL2TunnelPayloadInner,   /* Global Inner TPID of L2 Tunnel payload
*/
     bcmTpidTypeMimPayloadOuter,        /* Matched TPID of Transit MIM Payload
for Parser */
     bcmTpidTypeVxlanPayloadOuter,      /* Matched TPID of Transit VXLAN Payload
for Parser */
     bcmTpidTypeL2grePayloadOuter,      /* Matched TPID of Transit L2GRE Payload
for Parser */
     bcmTpidTypeVxlanPayloadInner,      /* Matched inner TPID of Transit VXLAN
Payload for Parser */
+    bcmTpidTypeItag = 11,               /* Global TPID value for I-Tag */
     bcmTpidTypeCount                   /* Always Last, not a usable value */
 } bcm_switch_tpid_type_t;
```

New members added to bcm_switch_match_service_t

```
typedef enum bcm_switch_match_service_e {
     bcmSwitchMatchServiceMiml = 0,      /* Mac-in-Mac Lite(MiML). Required
                                            fields: inner_ethtype, outer_ethtype.
*/
     bcmSwitchMatchServiceCustomHeader = 1, /* Custom Header. Required fields:
                                            value32, int_pri, color. */
     bcmSwitchMatchServiceGtp = 2,       /* GPRS Tunnel Protocol packets.
```

```
                                          Required fields: match_type,
                                          src_port, dst_port. */
+    bcmSwitchMatchServiceHsrPrpSupervision = 3, /* Seamless Redundancy HSR/PRP
+                                           supervision protocol. Required
+                                           fields: ethtype1, ethtype2, dst_mac,
+                                           dst_mac_mask */
+    bcmSwitchMatchServiceDot1cbSupervision = 4, /* Seamless Redundancy 802.1CB
+                                           supervision protocol. Required
+                                           fields: ethtype1, dst_mac,
+                                           dst_mac_mask */
+    bcmSwitchMatchServiceLinkLocal = 5, /* Link-local MAC addresses. Required
+                                           fields: dst_mac, dst_mac_mask */
+    bcmSwitchMatchService__Count = 6    /* Must be last. */
 } bcm_switch_match_service_t;
```

New members added to bcm_switch_match_config_info_t

```
typedef struct bcm_switch_match_config_info_s {
     uint32 value32;         /* 32-bit data value to match */
     uint16 inner_ethtype;   /* 16-bit Inner Ethertype to match */
     uint16 outer_ethtype;   /* 16-bit Outer Ethertype to match */
     uint8 int_pri;          /* Internal packet priority to set when matched */
     bcm_color_t color;      /* Packet Color to set when matched */
     uint8 match_criteria;   /* Match criteria for GPRS Tunnel Protocol packets
*/
     #define BCM_SWITCH_GTP_MATCH_DST_PORT         0x1
     #define BCM_SWITCH_GTP_MATCH_SRC_PORT         0x2
     #define BCM_SWITCH_GTP_MATCH_SRC_OR_DST_PORT  0x3
     #define BCM_SWITCH_GTP_MATCH_SRC_AND_DST_PORT 0x4

     uint16 dst_port;        /* Matched destination port */
     uint16 src_port;        /* Matched source port */
+    uint16 ethtype1;        /* 16-bit Ethertype to match */
+    uint16 ethtype2;        /* 16-bit Ethertype (alternate) to match */
+    bcm_mac_t dst_mac;      /* Destination MAC address to match */
+    bcm_mac_t dst_mac_mask; /* Mask for destination MAC address */
 } bcm_switch_match_config_info_t;
```

bcm_switch_tpid_class_get

Get the tpid profile from a given port or ingress vlan translation action ID.

SYNOPSIS

```
#include <bcm/switch.h>

int bcm_switch_tpid_class_get(
    int unit,
    bcm_switch_tpid_class_t *tpid_class);
```

PARAMETERS

unit	(IN) Unit number.
tpid_class	(IN/OUT) TPID class retrieval information.

DESCRIPTION

Get the tpid class from a given port or ingress vlan translation action ID. The retrieved tpid class can be used for configuring egress native vlan classification and other purposes.

TPID class retrieval information structure.

```
typedef struct bcm_switch_tpid_class_s {
    uint32 flags;                         /* BCM_SWITCH_TPID_CLASS_XXX, unused
now. */
    bcm_switch_tpid_class_match_t match; /* Match criteria for tpid_class_id */
    bcm_gport_t port;                     /* Port ID */
    int vlan_translation_action_id;       /* Translation action ID */
    uint32 tpid_class_id;                 /* TPID-profile */
} bcm_switch_tpid_class_t;
```

RETURNS

BCM_E_xxx

bcm_switch_tpid_class_t_init

Initialize a bcm_switch_tpid_class_t structure.

SYNOPSIS

```
#include <bcm/switch.h>

void
bcm_switch_tpid_class_t_init(bcm_switch_tpid_class_t *tpid_class);
```

PARAMETERS

tpid_class	(IN/OUT) TPID class retrieval information.

DESCRIPTION

Initializes a TPID class retrieval information structure to default values.

Whereas bcm_switch_tpid_class_t is defined as:

```
/* TPID class retrieval information structure. */
typedef struct bcm_switch_tpid_class_s {
    uint32 flags;                       /* BCM_SWITCH_TPID_CLASS_XXX, unused
now. */
    bcm_switch_tpid_class_match_t match; /* Match criteria for tpid_class_id */
    bcm_gport_t port;                   /* Port ID */
    int vlan_translation_action_id;     /* Translation action ID */
    uint32 tpid_class_id;               /* TPID-profile */
} bcm_switch_tpid_class_t;
```

And bcm_switch_tpid_class_match_t is defined as:

```
/* TPID class match type. */
typedef enum bcm_bcm_switch_tpid_class_match_e {
    bcmTpidClassMatchPort = 0,          /* tpid_class_id from port. */
    bcmTpidClassMatchIngVlanActionId = 1, /* tpid_class_id from Ingress vlan
                                             translation action id */
    bcmTpidClassMatch_count  = 2        /* Number of match criterion */
} bcm_switch_tpid_class_match_t;
```

RETURNS

Nothing

Section 6.11: UDF Resources Management

Table 6.16: Options to bcm_udf_create

flags	Description	
BCM_UDF_CREATE_O_FLEX_COUNTER	Hint to the API so the udf allocation is adjusted to be used by Flexible Counter.	New

Section 6.12: VXLAN

Table 6.17: VXLAN port flags

Name	Purpose	
BCM_VXLAN_PORT_UDP_SRC_PORT_RANGE_ENABLE	UDP source port range checking for Entropy	New
BCM_VXLAN_PORT_TUNNEL_INNER_VLAN_DELETE	Delete Tunnel Inner VLAN Tag	New
BCM_VXLAN_PORT_TUNNEL_QOS_MAP	Use QoS mapped priority for egress Tunnel packet priority control	New
BCM_VXLAN_PORT_TUNNEL_OUTER_VLAN_DELETE	Delete Tunnel Outer VLAN Tag	New
BCM_VXLAN_PORT_VPN_BASED	To specify the VP is per VPN based	New

New members added to bcm_vxlan_port_t

```
typedef struct bcm_vxlan_port_s {
     bcm_gport_t vxlan_port_id;          /* GPORT identifier. */
     uint32 flags;                       /* BCM_VXLAN_PORT_xxx. */
     uint32 if_class;                    /* Interface class ID. */
     uint16 int_pri;                     /* Internal Priority */
     uint8 pkt_pri;                      /* Packet Priority */
     uint8 pkt_cfi;                      /* Packet CFI */
     uint16 egress_service_tpid;         /* Service Vlan TPID Value */
     bcm_vlan_t egress_service_vlan;     /* SD-TAG Vlan ID. */
     uint16 mtu;                         /* MTU */
     bcm_gport_t match_port;             /* Match port / trunk */
     bcm_vxlan_port_match_t criteria;    /* Match criteria */
     bcm_vlan_t match_vlan;              /* Outer VLAN ID to match. */
     bcm_vlan_t match_inner_vlan;        /* Inner VLAN ID to match. */
     bcm_gport_t egress_tunnel_id;       /* Tunnel Initiator ID */
     bcm_gport_t match_tunnel_id;        /* Tunnel Terminator ID */
     bcm_if_t egress_if;                 /* VXLAN egress object. */
     bcm_switch_network_group_t network_group_id; /* Switch split horizon network
group ID this VXLAN port belongs to*/
     int qos_map_id;                     /* QoS Mapped priority */
     uint8 tunnel_pkt_pri;               /* Tunnel Packet Priority */
     uint8 tunnel_pkt_cfi;               /* Tunnel Packet CFI */
     uint32 vnid;                        /* VXLAN packet VNID. */
+    int qos_map_id;                     /* QoS Mapped priority */
+    uint8 tunnel_pkt_pri;               /* Tunnel Packet Priority */
+    uint8 tunnel_pkt_cfi;               /* Tunnel Packet CFI */
 } bcm_vxlan_port_t;
```

Section 7: 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. This data does not include DNX device results.

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 6.5.7, 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
BCM9COMX2XMC86 (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-6.5.7	***sdk-6.5.6***	***sdk-6.5.5***
golden	153	153	153
warmboot	3248	3032	3004
auth	17	17	17
bfd	112	89	86
bhh	150	133	126
chip	9	9	9
coe	579	579	579
cosq	773	768	709
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	1686	1470	1435
higigproxy	129	129	129
infra	114	114	114
ipfix	17	17	17
ipmc	114	114	114
l2	344	337	337
l2gre	33	33	33
l3	553	546	525
l3.alpm	550	550	512
link	26	26	26
mim	46	46	46
mirror	173	173	173
misc	21	21	20
mpls	541	541	483
multicast	29	29	29
niv	66	66	65
oam	397	391	377
pkt	44	44	44
port	466	465	398

proxy	38	38	37
ptp	115	115	115
qos	13	13	13
rate	21	21	21
rtag7	50	43	43
rx	25	25	25
ser	157	157	157
stack	117	117	117
stat	439	403	392
stg	42	42	42
switch	217	197	197
time	33	33	33
tlvMsg	13	13	13
trill	47	47	47
trunk	252	252	232
tunnel	133	133	133
subport	31	31	31
vlan	242	240	240
vxlan	219	219	208
wlan	17	17	17
Test Suite Total	12848	12285	11911

Section 7.4: API Test Results

In this release, all tested devices passed our DVAPI regressions with over 99.8% passing rate. No device specific results are shown unless there is a certain device which does not meet this quality metric. Prior release notes will show historical data for each device and the expectation going forward is that each subsequent SDK release will maintain or improve upon the last release quality.

Section 7.4.1: All Devices

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

	sdk-6.5.7	***sdk-6.5.6***	***sdk-6.5.5***
golden	0.06 %	0.04 %	0.1 %
warmboot	0.06 %	0.1 %	0.1 %
bcm.bfd	0.02 %	0.0 %	0.1 %
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	0.04 %	0.05 %	0.1 %
bcm.cosq	0.05 %	0.06 %	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	0.12 %	0.14 %	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 %	0.01 %	0.1 %
bcm.l2	0.03 %	0.03 %	0.1 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.02 %	0.0 %	0.1 %
bcm.l3.alpm	0.01 %	0.01 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.01 %	0.02 %	0.1 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.01 %	0.05 %	0.0 %
bcm.multicast	0.07 %	0.1 %	0.1 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.02 %	0.01 %	0.1 %

bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.34 %	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 %	0.1 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.01 %	0.0 %	0.0 %
bcm.stack	0.01 %	0.0 %	0.0 %
bcm.stat	0.06 %	0.08 %	0.1 %
bcm.stg	0.00 %	0.02 %	0.0 %
bcm.switch	0.00 %	0.02 %	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	0.05 %	0.13 %	0.1 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.16 %	0.16 %	0.0 %
bcm.vlan	0.01 %	0.01 %	0.1 %
bcm.vxlan	0.00 %	0.01 %	0.0 %

bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.2 %	0.2 %

Section 7.5: PHY Test Results

Starting with SDK-6.5.3, 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*
56960_A0	phy82381_100G	198	4.55%
56960_A0	phy82764_10G	198	1.01%
56860_A0	phy82780_10G	198	0.00%
56860_A0	phy82780_40G	198	0.00%
56860_A0	phy82792_100G	198	3.03%
56867_A1	phy82764_10G	198	1.01%
56867_A1	phy82764_40G	198	1.01%
56960_B0	phy82864_100G	198	3.03%
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	6.06%
56960_B0	phy82864_10G_alt	198	6.06%
56960_B0	phy82864_11G	198	3.03%
56960_B0	phy82864_11G_alt	198	3.03%
56960_B0	phy82864_42GPt	198	1.01%
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	4.04%
56960_B0	phy82864_40G2x20_alt	198	4.04%
56960_B0	phy82864_42G2x20	198	2.02%
56960_B0	phy82864_42G2x20_alt	198	3.03%
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	0.51%
56867_A1	phy82332_100G_gbox	198	3.54%
56867_A1	phy82332_106GPt	198	0.51%
56867_A1	phy82332_40G	198	1.52%
56867_A1	phy82332_42G	198	1.52%
56867_A1	phy82332_10G	198	6.57%

56867_A1	phy82332_11G	198	6.57%
56860_A1	phy82332_retimer_1G	198	3.03%
56860_A1	phy82332_retimer_10G	198	6.57%
56860_A1	phy82332_retimer_11G	198	7.07%
56860_A1	phy82332_retimer_40G	198	4.55%
56860_A1	phy82332_retimer_100G	198	4.55%
56765_A0	phy54140_1G_Copper	198	0.51%
56765_A0	phy54140_1G_Fiber	198	0.51%
56565_A0	phy84868_10G	198	1.01%
56960_B0	phy82864_40GMux	147	5.44%

Section 7.5.2: Interop External PHY

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	7.43%
56560_B0	XE_G40_84328_40G	122	4.92%
56560_B0	XE_MT_84757_10G	69	1.45%
56860_A1	XE_QD28_82780_40G	141	0.00%
56860_A1	XE_QD28_82780_10G	207	0.00%
56860_A1	XE_SESTO_82764_40G_MUX	107	7.48%
56860_A1	XE_SESTO_82764_40G_PT	174	0.00%
56860_A1	XE_SESTO_82764_10G_PT	162	0.00%

56860_A1	XE_DINO_82332_40G_AN	132	4.55%
56860_A1	CE_DINO_82332_100G_PT_L1_10	242	0.83%
56860_A1	HG_DINO_82332_106G_GB	186	0.00%
56860_A1	CE_DINO_82332_100G_PT	242	0.83%
56860_A1	CE_DINO_82332_100G_GB_L2_11	307	3.58%
56860_A1	HG_DINO_82332_11G	77	6.49%
56860_A1	HG_DINO_82332_42G	77	5.19%
56860_A1	CE_DINO_82332_100G_PT_L2_11	242	0.83%
56860_A1	HG_DINO_82332_106G_PT	186	0.00%
56860_A1	HG_DINO_82332_10G	39	7.69%
56860_A1	HG_DINO_82332_106G_GB_L0_9	186	0.00%
56860_A1	CE_DINO_82332_100G_GB_L1_10	307	3.58%
56860_A1	XE_DINO_82332_10G	280	3.57%
56860_A1	CE_DINO_82332_100G_GB	339	1.77%
56860_A1	XE_DINO_82332_40G	143	11.89%
56860_A1	HG_DINO_82332_11G_RETIMER	89	2.25%
56860_A1	CE_DINO_82332_100G_RETIMER	242	0.83%
56860_A1	XE_DINO_82332_10G_RETIMER	194	2.58%
56860_A1	GE_DINO_82332_1G_RETIMER	152	5.26%
56860_A1	XE_DINO_82332_40G_RETIMER	191	3.14%
56860_A1	XE_G40_84328_10G	148	4.05%
56860_A1	XE_G40_84328_40G	122	4.92%
56860_A1	XE_G28_82322_10G	148	1.35%
56860_A1	XE_G28_82322_40G	122	0.82%
56860_A1	XE_54210	38	0.00%
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	54	0.00%
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	162	8.02%
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_B0	CE_MADURA_82864_100G_RETIMER_ULL	191	6.28%
56960_B0	XE_MADURA_82864_25G_ALT_RETIMER	54	0.00%
56960_B0	XE_MADURA_82864_40G_DUAL_ALT_RETIMER	126	7.94%
56960_B0	XE_MADURA_82864_40G_PT_ALT_RETIMER	162	0.00%
56960_B0	XE_MADURA_82864_10G_RETIMER_ULL	162	0.00%
56960_B0	HG_MADURA_82864_42G_PT_ALT_RETIMER	90	0.00%
56960_B0	XE_MADURA_82864_10G_RETIMER	162	0.00%
56960_B0	HG_MADURA_82864_106G_ALT_RETIMER	48	0.00%
56960_B0	XE_MADURA_82864_50G_RETIMER	54	0.00%

56960_B0	HG_MADURA_82864_11G_ALT_RETIMER	90	0.00%
56960_B0	HG_MADURA_82864_106G_RETIMER	86	16.28%
56960_B0	CE_MADURA_82864_100G_ALT_RETIMER	216	0.00%
56960_B0	HG_MADURA_82864_11G_RETIMER	90	0.00%
56960_B0	XE_MADURA_82864_50G_ALT_RETIMER	54	0.00%
56960_B0	XE_MADURA_82864_25G_RETIMER	54	0.00%
56960_B0	HG_MADURA_82864_42G_PT_RETIMER	90	0.00%
56960_B0	HG_MADURA_82864_42G_DUAL_ALT_RETIMER	90	0.00%
56960_B0	XE_MADURA_82864_40G_MUX_RETIMER	162	0.00%
56960_B0	XE_MADURA_82864_40G_DUAL_RETIMER	138	5.80%
56960_B0	CE_MADURA_82864_100G_RETIMER	200	7.00%
56960_B0	XE_MADURA_82864_10G_ALT_RETIMER	162	0.00%
56960_B0	HG_MADURA_82864_42G_DUAL_RETIMER	66	12.12%
56960_B0	XE_MADURA_82864_40G_PT_RETIMER_ULL	153	0.00%
56960_B0	XE_MADURA_82864_40G_PT_RETIMER	162	0.00%
56960_A0	XE_MADURA_82864_40G_PT	93	0.0%

Loopback Suite

Switch Device	*Port Macro*	*Total Tests*	*% Fail*
56560_B0	HG_G40_84328_42G	60	13.33%
56560_B0	XE_G40_84328_10G	148	7.43%
56560_B0	XE_G40_84328_40G	122	4.92%
56560_B0	XE_MT_84757_10G	69	1.45%
56860_A1	XE_QD28_82780_40G	33	3.03%
56860_A1	XE_QD28_82780_10G	55	0.00%

56860_A1	XE_SESTO_82764_40G_MUX	25	80.00%
56860_A1	XE_SESTO_82764_40G_PT	30	0.00%
56860_A1	XE_SESTO_82764_10G_PT	30	0.00%
56860_A1	XE_DINO_82332_40G_AN	5	100.00%
56860_A1	CE_DINO_82332_100G_PT_L1_10	56	0.00%
56860_A1	HG_DINO_82332_106G_GB	31	0.00%
56860_A1	CE_DINO_82332_100G_PT	56	0.00%
56860_A1	CE_DINO_82332_100G_GB_L2_11	63	0.00%
56860_A1	HG_DINO_82332_11G	15	66.67%
56860_A1	HG_DINO_82332_42G	15	73.33%
56860_A1	CE_DINO_82332_100G_PT_L2_11	56	0.00%
56860_A1	HG_DINO_82332_106G_PT	31	0.00%
56860_A1	HG_DINO_82332_10G	10	100.00%
56860_A1	HG_DINO_82332_106G_GB_L0_9	31	0.00%
56860_A1	CE_DINO_82332_100G_GB_L1_10	63	0.00%
56860_A1	XE_DINO_82332_10G	40	87.50%
56860_A1	CE_DINO_82332_100G_GB	63	0.00%
56860_A1	XE_DINO_82332_40G	35	85.71%
56860_A1	HG_DINO_82332_11G_RETIMER	20	0.00%
56860_A1	CE_DINO_82332_100G_RETIMER	56	0.00%
56860_A1	XE_DINO_82332_10G_RETIMER	40	0.00%
56860_A1	GE_DINO_82332_1G_RETIMER	39	58.97%
56860_A1	XE_DINO_82332_40G_RETIMER	35	0.00%
56860_A1	HG_G40_84328_42G	10	0.00%
56860_A1	XE_G40_84328_10G	10	0.00%
56860_A1	XE_G40_84328_40G	8	0.00%
56860_A1	XE_G28_82322_10G	10	0.00%
56860_A1	XE_G28_82322_40G	8	0.00%
56860_A1	XE_54210	8	0.00%

56960_B0	HG_MADURA_82864_42G_PT	66	6.06%
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	3.03%
56960_B0	XE_MADURA_82864_25G	22	27.27%
56960_B0	HG_MADURA_82864_42G_PT_ALT	66	0.00%
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	0.00%
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	43.18%
56960_B0	CE_MADURA_82864_100G	88	31.82%
56960_B0	XE_MADURA_82864_40G_PT	66	0.00%
56960_B0	CE_MADURA_82864_100G_RETIMER_ULL	77	46.75%
56960_B0	XE_MADURA_82864_25G_ALT_RETIMER	22	4.55%
56960_B0	XE_MADURA_82864_40G_DUAL_ALT_RETIMER	66	36.36%
56960_B0	XE_MADURA_82864_40G_PT_ALT_RETIMER	66	0.00%
56960_B0	XE_MADURA_82864_10G_RETIMER_ULL	66	27.27%

56960_B0	HG_MADURA_82864_42G_PT_ALT_RETIMER	66	0.00%
56960_B0	XE_MADURA_82864_10G_RETIMER	66	0.00%
56960_B0	XE_MADURA_82864_50G_RETIMER	22	0.00%
56960_B0	HG_MADURA_82864_11G_ALT_RETIMER	66	0.00%
56960_B0	HG_MADURA_82864_106G_RETIMER	88	28.41%
56960_B0	CE_MADURA_82864_100G_ALT_RETIMER	88	2.27%
56960_B0	HG_MADURA_82864_11G_RETIMER	66	0.00%
56960_B0	XE_MADURA_82864_50G_ALT_RETIMER	22	0.00%
56960_B0	XE_MADURA_82864_25G_RETIMER	22	4.55%
56960_B0	HG_MADURA_82864_42G_PT_RETIMER	66	0.00%
56960_B0	HG_MADURA_82864_42G_DUAL_ALT_RETIMER	66	0.00%
56960_B0	XE_MADURA_82864_40G_MUX_RETIMER	66	36.36%
56960_B0	XE_MADURA_82864_40G_DUAL_RETIMER	66	24.24%
56960_B0	CE_MADURA_82864_100G_RETIMER	88	18.18%
56960_B0	XE_MADURA_82864_10G_ALT_RETIMER	66	0.00%
56960_B0	HG_MADURA_82864_42G_DUAL_RETIMER	66	24.24%
56960_B0	XE_MADURA_82864_40G_PT_RETIMER_ULL	55	0.00%
56960_B0	XE_MADURA_82864_40G_PT_RETIMER	66	0.00%
56960_A0	XE_MADURA_82864_40G_PT	33	15.2%

Section 7.5.3: Interop Internal PHY

P2P Suite

Switch Device	*Port Macro*	*Total Tests*	*% Fail*
56960_B1	HG_64XD53	108	0.00%
56960_B1	HG_32X106	72	0.00%
56960_B1	HG_128X27	72	0.00%
56960_B1	XE_32X100	136	2.94%
56960_B1	XE_64XD50	204	2.94%
56960_B1	XE_128X25	136	2.94%
56860_A1	HG_32X42	104	1.92%
56860_A1	XE_32X40	132	4.55%
56860_A1	HG_8x100_343	26	7.69%
56860_A1	CE_8X100_343_IEEE	59	8.47%
56860_A1	HG_104x10	36	0.00%
56860_A1	XE_104x10	101	1.98%
56860_A1	CE_8x100_442	59	8.47%
56260_A0	GE_VIPER_12x2P5	36	8.33%
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	0.00%
56160_A0	SGMII_GPHY	48	0.00%
56160_A0	QSGMII_GPHY	48	0.00%
56565_A0	XE_MAKO	16	37.50%

56960_A0	XE_128X25	89	24.7%

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	27.27%
56860_A1	HG_8x100_343	11	0.00%
56860_A1	CE_8X100_343_IEEE	11	0.00%
56860_A1	HG_104x10	11	0.00%
56860_A1	XE_104x10	22	27.27%
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	3.03%
56160_A0	SGMII_QTC	40	0.00%

56160_A0	SGMII_GPHY	20	0.00%
56160_A0	QSGMII_GPHY	20	0.00%
56960_A0	XE_128X25	22	0.0%

Section 7.6: 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 we have been working down the backlog of issues. Below shows the current baseline and progress in recent 6.5.x releases:

Section 7.6.1: Unresolved Static Code Analysis Issues

Line of Business	***New baseline as of 3/1/16***	***Open Issues SDK 6.5.7***	***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	47	63	124	107	104	61	75	125
XGS	90	10	15	23	37	86	16	27	23
SBX	40	*	*	*	*	34	1	0	48
SerDes	35	10	38	25	23	34	12	45	45
Common	45	4	27	32	42	42	11	16	46
Total	330	71	143	204	209	300	101	164	301

* As of SDK 6.5.4 we are no longer tracking SBX quality issues as part of our 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 5: Resolved Service Impacting Defects

Reference	*Chips*	*Affected Versions*	*Errata Synopsis*	*Details*
SDK-110589	56960 , 56965	6.5.6, 6.5.5, 6.5.4, 6.5.3, 6.4.11, 6.4.10, 6.4.9, 6.4.8, 6.4.7	Lookup failure causes DLF due to L2 HW learning MAC address issue	Read to L2_ENTRY_ONLY_TILE or L2_ENTRY_TILE table can corrupt the LP table when L2_MGMT is active, resulting in known unicast traffic cannot hit the corresponding L2 entry and flood as DLF. During warmboot, L2_ENTRY_TILE table is read to rebuild cache, which causes this issue.

Section 9: Resolved Issues for 6.5.7

Section 9.1: Resolved Improvements

The table below lists improvement JIRAs that have been added to SDK 6.5.7:

Number	*CSP*	*Chips*	*Release Notes For 6.5.7*
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-100807	1057155	88670_A0	In Egress PMF, it was not supported to reach the Ethertype field. Now a new qualifier is added, bcmFieldQualifyEthernetData, that can be used only as a base for a data qualifier and only in Egress. It contains 80 bits of Ethernet data (without MAC), and the Ethertype can be obtained from it at offsets 0, 32 or 64 depending on the number of VLAN tags in the packet passed to the Egress.
SDK-101159	1050384	88670_A0 88670_B0 88675_A0 88675_B0	In Ingress PMF, new preselector is added bcmFieldQualifyVPortRangeCheck, which allows to qualify packets according to range of Out-LIFs

SDK-101456 1055852	88675_A0 88675_B0	Background: When the outgoing packet needs to add GRE tunnel type, it is required to set the ether-type/next protocol (16 bit) of the header above the GRE inside the GRE header. This is automatically covered for Eth/IPv4/IPv6 headers, for other types of headers a default value is used. This default value is taken from the register: IPV4_UNKNOWN_HEADER_CODE_ETHERNET_TYPE. Change: Added support for GRE over ERSPAN - in this case the GRE next protocol is set to (0x88BE) regardless of the default value in the register.
SDK-103736 1072336	88470_A0	The user can now provide the desired trap strength for the trap destination given via the trap strength encoded in bcm_oam_endpoint_info_t->remote_gport variable during endpoint creation or in destination parameter of bcm_oam_endpoint_action_t while configuring bcmOAMActionUcFwd or bcmOAMActionMcFwd actions. Also, new soc properties oam_trap_strength_passive and oam_trap_strength_level have been added for the user to specify the passive error and level trap error strengths.
SDK-104598 1068970	88670_B0	When BFD tunneltype is bcmBFDTunnelTypeMplsTpCc and egress_label is 13 (GAL label), the OAMP will generate only the GAL label with no additional LSP label.
SDK-104900 1075070	88675_A0 88675_B0	The IPv4 DC feature with 24bit forwarding is enabled by setting the SOC property custom_feature_ext_tcam_dc_24bit_result. It allows the use of 4 ACL results with IPv4 DC instead of the 3 available with the IPv4 DC 48bit forwarding.
SDK-104978 1062316	88670_A0 88670_B0	Virtual traps "bcmRxTrapsL2Cache" can now be created with BCM_RX_TRAP_WITH_ID flag. i.e

		bcm_rx_trap_type_create(unit,BCM_RX_TRAP_WITH_ID,bcmRxTrapL2Cahce,&trap_id) trap_id will hold the desired virtual trap to create (Reserved MC trap or General Programable trap)
SDK-105764 1070433	88670_A0 88670_B0	In external (KBP) ACLs, for IPv6 added support to create ACLs with the DIP and SIP of the IPv6 header also when FWD is not done in the external TCAM. until this feature added there was no possibility to create does type of ACLs because of the limitation of 160bit key size. This issue was solved by using the DIP and the SIP for the Packet header (doesn't require copy engines),
SDK-106221 1068973	88670_A0 88670_B0	The BFD endpoint info's remote_gport variable can be used to represent the trap strength that should be assigned to the endpoint's trap action apart from the trap code associated with it.
SDK-106224 1081290	56850_A0 56850_A1 56850_A2 56960_A0 56970_A0	In previous SDK releases: flex counters with UDF pkt attribute is not supported. This is supported now.
SDK-106563	88670_A0	L3 Interface: A new internal field routing-enablers-bm-RIF-profile of L3 Intf (In-RIF) is exposed. routing-enablers-bm-RIF-profile is part of the full RIF-profile namespace (Interface-class). The number of bits routing-enablers-bm-RIF-profile is determined according to SOC-property number_of_inrif_mac_termination_combinations. The field determines each interfaces are disable for a certain In-RIF e.g. In-RIF 5 disabled interface is IPv6 UC, MPLS. Field name in L3 interface is intf_class_route_disable. The new field can used by the Field-Processor as part of bcm_field_qualify_IncomingIpIfClass to qualify a certain packets.

		An example of use is for Explicit-NULL case. If L3 interface is mpls-enabled, packets with explicit-null label still get forwarded even it is disabled. Now, we can use API bcm_l3_ingress_get() to get routing-enablers bit map (intf_class_route_disable), and then call PMF qualify to forward or drop the packets.
SDK-106575 1081313	88470_A0	New feature: Ingress Scheduler compensation can be specified for input PP port. Details can be found in UM.
SDK-107692 1090031	56860_A0 56860_A1	In previous release SDK does not show counters of flexible assigned queues in TD2+ in diag command "show counters". This is supported now.
SDK-107836 1090394	56860_A0 56860_A1	In previous release SDK does not show counters of flexible assigned queues in TD2+ in diag command "show portmap". This is supported now.
SDK-108177 1008283	88670_A0 88670_B0	Terminate MPLS/VxLAN/GRE/802.1BR tunnels based on source port, and not based on TT classification.
SDK-108210 1091213	56340_A0 56340M_A0 56760_A0 56760_B0	Support for bcmCosqControlQgroupMinEnable added for Helix4 device.
SDK-108675 1092586	88670_A0 88670_B0	VPWS tagged mode PWE2AC direction: When using bcm api bcm_flexible_cross_connect is set for same lif with both BCM_PORT_MATCH_PORT_VLAN_STACKED and BCM_PORT_MATCH_PORT_VLAN, native ETH double tagged packets will now looked up by 2 databases: single-VLAN and double-VLAN. In case both result will match: double tagged result will be prioritized. In previous versions, there was an assumption that double -VLAN and single-VLAN databases couldn't match on the same double-tagged packets.
SDK-108870 1094873	56850_A0 56850_A1 56850_A2 56860_A0	In previous releases, there was no way to show ALPM brief debug information. In this release, this has been fixed by providing a BCM shell command ‘l3 alpm show brief’.
SDK-108965 1094190	88670_B0	Which actions/scenarios may trigger the issue: KNET filter didn't support to match trap qualifier.

		What is the impact when the issue occurs: 1. pmf rules which share the same trap id couldn't be distinguished 2. there may be user-defined trap qualifer, but knet couldn't match it. What is the impact of the fix: the trap id and trap qualifer are in following format: trap_id << 16 | trap_qualifier knet will fetch both trap_id and trap_qualifer from FHEI headers. and compare them with the value in knet filter.
SDK-109094	88375_A0 88375_B0 88670_A0 88670_B0	PON: In the previous release, SDK doesn't support 16bits tunnel ID and 64 PON ports. A problem occurs when customer want to use 16bits tunnel ID and 64 PON ports. The changes is to support the function when pon_pp_port_mapping_bypass=1.
SDK-109424 1093417	88270_A0 88470_A0 88470_B0	The transmit of CCM packets by HW acceleration engine can be stopped (without disturbing the RX of CCM packets) by setting CCM period to zero during OAM endpoint creation for accelerated endpoints. This is true for Qumran AX (88470) and Qumran UX (88270) platforms only. For Jericho (88670) and previous platforms, the way to disable CCM packet transmission is to set the soc property custom_feature_oam_ccm_rx_wo_tx=1 and also setting CCM period to zero during OAM endpoint create. The soc property is not required for QAX and will cause init error if used.
SDK-110004 1095408	88670_A0	Added validation check for encap_id.
SDK-110090 1097071	88675_B0	Which actions/scenarios may trigger the issue: steps as below: 1. add mirror configuration to mirror packets to CPU port 2. create knet filter to match bcmRxReasonMirror 3. tx packets and verify. only mirror packets are supposed to be sent to CPU. What is the impact when the issue occurs: knet filter doesn't work as expected. not only the mirrored packets but also all other packets to CPU are sent to CPU. so filter didn't take effect.

		What is the impact of the fix: 1. bcmRxReasonMirror to match mirrored packets 2. bcmRxReasonSampleSource to match snooped packets
SDK-110157 1096072	56160_A0 56160_B0	In bcm_multicast_egress_xxx() APIs, change the error return from BCM_E_UNAVAIL to BCM_E_NOT_FOUND for MC group not existed condition.
SDK-110253 1066493	56260_A0 56260_B0	Increase max linkphy subports allowed per linkphy physical port from 4 to 16
SDK-110297 1098216	56860_A0 56860_A1	Add bcm_tx support for sending PPD2 packet with vxlan vp.
SDK-110431 1097290	88670_B0	The diagnostics command, located in diag_oam.c has a restriction for reading only standard Opcode values (1-55), and is not intended to be able to read the UDF opcode which is 62 (0x3e). It no longer has a restriction for such opcodes and will not produce an error.
SDK-110615 1099840	56160_A0 56160_B0	Added support of flag BCM_EXTENDER_FORWARD_COPY_TO_CPU for BCM56160.
SDK-110725 1089421	88670_B0 88675_B0	Add support for MACSEC in DNX dispatch layer in the current release.
SDK-110984 1098558	56850_A0 56850_A1 56850_A2	In previous releases, the HW L3_IPMC table would be written again by the API bcm_ipmc_add even if there was no change on related IPMC entry. In this release, this has been fixed.

Section 9.2: Resolved Issues

The table below lists all defects resolved in SDK 6.5.7:

Number	*CSP*	*Chips*	*Release Notes For 6.5.7*
SDK-93232	1021618	88670_B0	PLL configuration has been removed from furia_phy_autoneg_ability_set to fix the 25G link issue with Jericho + Furia
SDK-99066		56260_B0	Viper does not have Ucode and no DSC -(Digital Conditioning Block) as a result dsc commands are not supported
SDK-99496	1046255	88670_A0	When speculative parsing was used with BOS search on a IPv4/6oMPLS(1-3 labels)oEth packets the configured LB preformed hashing on unexpected parts of the headers. The issue still exist, and now BOS-ELI search is non-functional. Will be available in 6.5.8 under new LB-scheme
SDK-100740	1042750	AllChips	Add hard reset printf for FE3200 SER.
SDK-102707	1044362	56860_A0 56860_A1	Fixed dynamic probing of 1G Ext PHY by correcting the speed value based on config property
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-105245		56846_A1	Add a check for soc in soc_linkscan_pause/continue to prevent system from crashing.
SDK-105459	1069874	88670_A0 88670_B0	the SOC property total_ram_size can be set to represent actual total Dram size. there's no longer the need to round it down to the nearest thousand. the calculation for compensating on memory needed

			for FBC and memory rendered unusable due to OCB limitations are done internally. this Fix is required when using big dram_buffer sizes and in the same time small size of total_ram_present in order to not be exposed to Dram CRCs, for more information check relevant KM article on the issue. it should be noted that under worst case scenarios actual total Dram size might be smaller by ~138MB
SDK-106071	1081629	56960_A0	In previous releases, dump command didn't return an error if mem index was out of range. In this release, dump command will return an error if mem index is out of range
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-106421	1083106	88670_A0	Which actions/scenarios may trigger the issue: bcm_petra_l2_cache_profile_set to set register IHP_MEF_L_2_CP_DROP_BITMAP on X86 CPU What is the impact when the issue occurs: the high 32bits and low 32bits were swapped wrongly on X86 CPU BCM.0> getreg IHP_MEF_L_2_CP_DROP_BITMAP IHP_MEF_L_2_CP_DROP_BITMAP.IHP0[0x2db]=0x4924924992492492: <MEF_L_2_CP_DROP_BITMAP=0x4924924992492492> IHP_MEF_L_2_CP_DROP_BITMAP.IHP1[0x2db]=0x4924924992492492: <MEF_L_2_CP_DROP_BITMAP=0x4924924992492492> it should be: BCM.0> getreg IHP_MEF_L_2_CP_DROP_BITMAP IHP_MEF_L_2_CP_DROP_BITMAP.IHP0[0x2db]=0x

			9249249249249249: <MEF_L_2_CP_DROP_BITMAP=0x9249249249249249> What is the impact of the fix: bcm_petra_l2_cache_profile_set couldn't work correctly
SDK-106595		5604X 56340M 56342M 56344M 5634X 5645X 56548 56547 5684X	Changed the shadow register setting which will be used during speed transition for single lane mode in warpcore B. The speed configs for this shadow register is changed from 10G to 1G during init.
SDK-106616	1085378	56860_A0 56860_A1	SOC_EGRESS_METERING_LOCK must be taken before removing shaper setting on the LLS nodes,and the lock must be released after restoring the setting. Also PORT_LOCK shouldn't be taken in the cosq module.
SDK-106910	1086636	88660_A0 88670_A0 88670_B0	OAM: Bailout exit condition from interrupt handler was added. In case of 1024 consecutive interrupts, user callback functions are called and the buffer gets cleared.
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 for now as we think it should be enough. Any increase in that number will affect scaling in future.

SDK-106958	1080605	88670_A0 88950_A0	Fixed a bug in diag shell soft reset command. When running soft reset which involves fabric interface reset multiple times on a FAP device (e.g. Jericho) connected to a FE3200, using one of these commands: devicereset 0x100 4 devicereset 0x200 4 devicereset 0x400 4 there may be continuous drops of cells in the FE3200, after the soft resets are done. The issue was fixed.
SDK-107115	1086466	56860_A0 56860_A1	Maintaining New Logical port to IFP stream ID mapping to provide unique stream IDs for each logical port, so that it retains the same stream ID during flex operation for a valid port configured in IFP entries.
SDK-107200	1082637	56545_A0 56545_A1 56545_B0	The earlier SDK WAR did not take care of the case where stacking systems had trunk ports. Incorrect accounting occurred since no entries were removed from MODPORT_MAP_SW or any other table during port shut. The fix was to first identify the trunk group the port belong to during a link down event and then remove it from the trunk group. When the next link up event occurred, the port was restored to the trunk group.
SDK-107201	1088070	56960_A0 56960_B0 56960_B1	added the check for interface /speed set, if no change, then tx taps will not be re-programmed.
SDK-107242	1086599	88375_A0 88375_B0 88470_A0	When using bcm_cosq_gport_discard_set() the drop probability that is written to the HW might be wrong for some drop probabilities. When using bcm_cosq_gport_discard_get() the drop probability that read from the HW might be very inaccurate for some drop probabilities. Fixed.
SDK-107350	1087121	56260 5645X	multicast service queue allocation logic corrected.
SDK-107407	1088952	56960_A0	In previous releases, There is a deadlock risk between LC_LOCK with LC_HANDLE_LOCK if the

		56960_B0 56960_B1	customer invokes APIs in the callback of bcm_linkscan_register. Meanwhile, The guide of SDK does not clarify that the customer must not invoke any API in the callback of bcm_linkscan_register that may result in a deadlock. The issue has been fixed in this release.
SDK-107432	1088887	88375_A0	OAM: If user call bcm_oam_endpoint_create() with a invalid intf_id, which will cause SDK process quit out. This issue affects OAM mep creating .After this fix, bcm_oam_endpoint_create() will return BCM_E_PARAM if using invalid intf_id and user can get prompt information.
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-107478	1087983	88375_A0 88670_A0	in previous version, adding dynamic port after warm-boot may cause egress interface override, resulting in traffic loss. Fixed,
SDK-107481	1088395	56460_A0 56460_B0 AllChips	Fixed by proper recognition of odd/even counters and remove accordingly.
SDK-107489	1083020	88670_B0	OAM: Memory leakage found on bcm_oam_init(). When API is called multiple times (on reload) API failed to allocate memory. Memory leakage was found and fixed.
SDK-107568	1088222	56540_A0 56540_B0	LINK_STATUSr register was not getting updated correctly. Due to this Traffic was not getting switched to original Hg port on enabling hg trunk failover. As on enabling Hg trunk failover, H/w checks LINK_STATUSr to determine whether to send traffic to original port or failover port. Corrected the LINK_STATUSr update.

SDK-107584	1088381	56460_A0 56460_B0	On KTX We use S1 regs in hardware for COE but use S0 node in software. So instaed of acceing S0 indices in hardware we shall access S1 regs.
SDK-107586	1068016	88375_A0 88375_B0 88650_A0 88650_B0 88660_A0 88675_A0 88675_B0	In case of an error during init sequence, segmentation fault may happen as a result of accessing uninitialized mutex. Fix was made so that instead of a crash, an error message is printed out.
SDK-107608	1084879	56640_A0 56640_A1 56640_B0	Problem: When service_queue_dynamic_config config property is set, the logic to identify whether the classifier is used, during service port map clear, by other ports was incorrect. Solution: The logic is fixed to accommodate offsets that are greater than max offset. When the offset exceeds the max value, HW calculates the offset after a wrap around.
SDK-107615	1069184	84757_C0	Revert the change made in SDK-87375. The fix for SDK-87375 is not required since FC line side link will go down if system side link is down.
SDK-107620	1088900	53443_B0	fix problem about IPG=0 while runtime config property to change valid ports is applied.
SDK-107665	1089874	56860_A0 56860_A1	In previous releases, API bcm_port_ability_remote_get didn't get 1000M data ability for CL37 AN. This issue has been fixed in this release.
SDK-107679	1089154	56640_A0 56640_A1 56640_B0	Corrected cosq control behavior for bcmCosqControlEgressMCSharedDynamicEnable and bcmCosqControlEgressUCSharedDynamicEnable in TR3 devices
SDK-107680	1088913	56460_A0 56460_B0	EXP will be taken from delay_info pkt_pri in SOBMH mode.
SDK-107688	1086093	AllChips	In previous releases, bcmSwitchL2CpuAddEvent/bcmSwitchL2CpuDeleteEvent switch controls didn't support tomahawk. In this release, bcmSwitchL2CpuAddEvent/bcmSwitchL2CpuDeleteEvent switch controls do support tomahawk.

SDK-107689	1088867	56450_A0 56450_B0 56450_B1	MMU_MC_REDIRECTION_PTR count in L3_ipmc table was not decremented after calling bcm_multicast_egress_subscriber_delete. pass 0 instead of ipmc_id to _bcm_kt_ipmc_mmu_mc_remap_ptr inf function _bcm_kt_ipmc_set_remap_group
SDK-107712	1087223	56450_A0 56450_B0 56450_B1 AllChips	"soc_linkscan_pause" is called on initiating a MIIM transaction and this function disables the hardware linkscan ports, this can result in some link state change events to be lost, this is now fixed by not disabling these.
SDK-107714	1089818	56850_A0 56850_A1 56850_A2	Internal Dirty flag need to be updated per action during entry remove.
SDK-107721	1090242	56850_A0 56850_A1 56850_A2 56860_A0 56960_A0	In previous releases, when called API bcm_l3_host_add to replace an old ECMP group by a new one which had some members in common, it wouldn't copy the resilient hashing flow set entries containing the shared members from the old ECMP group to the new ECMP group. This has been improved in this release to reduce flow-to-member reassignments.
SDK-107731	1088776	88375_A0 88375_B0	UP MEP packet can be trapped to CPU type port. A problem occurs when port vlan COE feature is enabled on UP MEP. Packet trapped to CPU type port didn't have correct system header. Improvement for COE port is added to resolve this issue.
SDK-107777	1082627	88670_B0	Increase the your_disc_ref_counter in end of function(BFD create) instead of current location.
SDK-107780	1090554	56960_A0 56960_B0 56960_B1	In previous releases, bcm_vlan_gport_get_all didn't get all virtual ports belong to a specific vlan on Tomahawk. In this release, bcm_vlan_gport_get_all is able to get all virtual ports belong to a specific vlan on Tomahawk
SDK-107789	1081731	88375_B0 88670_B0	PortVlan COE and Higig will co-exist if they are enabled on different ports. A problem occurs when Higig ports exist with PortVlan COE. Packet received from COE port will parse incorrectly.

			Improvement for COE port is added to resolve this issue.
SDK-107792	1087715	AllChips	In previous release, while calling bcm_vlan_member_set API for multiple ports, vp group lock was grabbed and caused deadlock since unlocking vp group lock was missed. This has been fixed in this release.
SDK-107866	1089467	56860_A0 56860_A1	In previous releases, PHY84858 name and link status were wrong after the port initialization. This issue has been fixed in this release.
SDK-107871	1074428	AllChips	In the previous release, arad_pp_signal_mngr_signal_get() used the wrong core id to get the signal. In this release, this issue has been fixed.
SDK-107888	1085185	56850_A0 56850_A1 56850_A2	In previous releases, when created an egress object with BCM_L3_VXLAN_ONLY flag and then called command "l3 egress show", the VLAN id printed was incorrect. This has been fixed in this release.
SDK-107896	1083094	88670_A0 88670_B0 88675_A0 88675_B0	While trying do disable snoop port configuration using bcm_port_sample_trap_set() the snoop was not disabled. This is Fixed.
SDK-107897	1090119	56860_A0 56860_A1	In the previous releases, match count in VP was not cleared completely when mpls failover port was deleted. In this release, the issue is fixed.
SDK-107954		56850_A2	In previous releases, L3 intf created internally by FP API is not retained after WB under certain circumstances. This has been fixed in this release.
SDK-107966	1090803	88950_a0	Fix initialization for devices with repaired memories. Prior to this fix, some devices with memories repaired as part of the production process could fail upon initialization.
SDK-107969	1088636	56860_A0 56860_A1	In previous releases, DSCP profile reference count were not updated correctly in api bcm_qos_port_map_set. This has been fixed in this release.
SDK-107970	1091131	56850_A0 56850_A1 56850_A2	In previous releases, when trying to use bcm_mirror_port_dest_delete() to maintain ingress mirroring on an NIV port whose associated egress object being removed, it would fail with E_NOT_FOUND. This has been addressed in this release.

SDK-107971	1089032	AllChips	In the previous releases, the parser of config.bcm always expects two ":" delimiters before EQ. In this release, only one ":" delimiter is expected before EQ.
SDK-108022	1089838	56850_A0 56850_A1 56850_A2	Fixed by checking all the parts of the entry in bcm_field_action_get.
SDK-108064	1091282	88670_B0	Turning on MSI in the CMIC causing invalid message corrupt host CPU memory. If LDK CPU don't support MSI or BDE don't support MSI, the MSI in the CMIC will be turned off .
SDK-108089	1087664	56850_A0 56850_A1 56850_A2	The flex stat written for a byte mode stat counter should be 64 bits where as only 32 bitsdata is initialized. Allocating two 32 bit data to set the stat counter of byte type.
SDK-108147	1090737	88670_A0 88670_B0	Speculative field after the MPLS can be used now for hashing but BOS/ELI support (with speculative) was removed. Support in both cases will be added in version 6.5.8
SDK-108149	1091277	56960_A0 56960_B0 56960_B1	Fixed the crash in "fp show" when compiled with DNX and ESW.
SDK-108158		88650_A0 88660_A0 88670_A0 88750_A0 88950_a0	When Hard Reset is required by critical SER interrupt, SDK does soft reset action actually, the hard reset action called or not will decided by customer via callback function, simultaneously SDK will dump out "soft reset" info in the previous release, but this info maybe misleading. In this release, the info has been changed to "Hard Reset is Required".
SDK-108166	1091027	56460_A0 56460_B0	On calling bcm_mpls_port_add() with BCM_MPLS_PORT_REPLACE flag and trying to replace Trunk Id with same Trunk Id, original SVP linkage from SOURCE_TRUNK_MAPm was getting deleted, same has been fixed to ensure that SVP linkage is retained in this case.
SDK-108242	1092196	88670_B0	Add the lif profile info to "diag pp LIF "
SDK-108257	1092279	56640_A0 56640_A1 56640_B0	In previous releases, the eyescan was broken. In this release, this issue has been fixed.
SDK-108316	1090557	88660_A0	While setting BFD_SINGLE_HOP over IPv6, IPv4 trap were triggered. This is now fixed
SDK-108327	1091459	88470_A0	OAM: OAMP TX machine configuration was not optimized for low clock rates. LMM packets were not correctly generated. This is now handled internally.

SDK-108356	1092647	56640_A0 56640_A1 56640_B0	In previous releases, the Veye only ran on 4 lanes for 100G port on Warpcore C. In this release, this issue has been fixed.
SDK-108357	1092622	88470_A0 88670_A0 88680_A0	Slow ports with jumbo packets might experience reassembly timeout error. Reassembly timeout value adjusted.
SDK-108401	1092007	56640_A0 56640_A1 56640_B0	Shared pool resume limit was not getting updated when shared pool limit is updated, same has been fixed.
SDK-108472	1092918	56846_A0	Some mmu memories (like MMU_THDO_CONFIG_0,MMU_THDO_OFFSET_0, etc) have several physical copies. When perform CPU writing to these memories, the corresponding physical copies' cache is updated together now.
SDK-108532	1082822	88375_A0 88375_B0	Set/delete internal functions for CFM MAC setting handles error scenario of Multicast mac and returns error. This will now result in bcm_l2_station_add API returning error when a multicast mac is passed in the bcm_l2_station_t.dst_mac parameter of the API with flags parameter set to BCM_L2_STATION_OAM.
SDK-108536	1086834	88375_A0 88375_B0	in previous version, CGM default Thresholds configuration may not be configured as expected when dynamic_port is enabled. SDK map TC0-TC7 to pool 0, but allocate buffers and descriptor only to TC0-TC3. Fixed.
SDK-108641	1093665	56640_A0 56640_A1 56640_B0	Endpoint indexes are being updated in endpoint config during wamboot recovery.
SDK-108647		88670_A0	SCH_TOKEN_MEMORY_CONTROLLER_TMC was cached by SW even though it is a dynamic memory. It was causing incorrect SER handling. In this release this table has been removed from the cache.
SDK-108701	1091490	56850_A0 56850_A1 56850_A2	In previous releases, the usage information of bucket could not be recovered successfully during warmboot, which resulted in the failure of ALPM sanity check. In this release, this has been fixed.
SDK-108738	1092674	88675_A0 88675_B0	Changed field adjust_factor_bds in structure SOC_TMC_ITM_FADT_DROP_INFO from uint8 type to int8 type in order to permit using negative numbers.

SDK-108815	1089451	88670_A0	Allow to create VSQ-E for recycle and CPU ports. Such VSQ can't be used for flow control generation (only for congestion management).
SDK-108823	1093807	88680_A0	When working with ILKN, it is always recommended to work with ILKN lane map SOC property. However, the specific case of ILKN 7 lanes where there are no swaps on the board, is now supported without setting ILKN lane map SOC property.
SDK-108856	1094706	56960_A0 56960_B0 56960_B1	In previous releases, bcm_mpls_port_add with replace flag overwrited the DISABLE_VLAN_CHECKS field. In this release, bcm_mpls_port_add with replace flag will not overwrite the DISABLE_VLAN_CHECKS field.
SDK-108867	1094521	56640_A0 56640_A1 56640_B0	The unlock sequence for L2X memory is modified for BCM56640 devices.
SDK-108881		56860_A0 56860_A1	In previous releases, Portmod didn't implement SGMII AN setting. This issue has been fixed in this release.
SDK-108891	1059140	56850_A0 56850_A1 56850_A2	There is something wrong in pw rx packet print. In pw_dump_packet_rx, SDK uses pkt->flags instead of stk_flags to print src_gport/dst_gport info. Now this issue has been fixed.
SDK-108901	1092114	88670_A0 88670_B0	In previous release for EM table 1b-ECC recovery, EM management unit was not disabled, this maybe cause unexpected result, because the EM table can be updated during the EM ECC error recovery. In this release, EM management unit was disabled in the period of EM SER.
SDK-108908	1094054	56850_A0 56850_A1 56850_A2	In previous releases, When command "link off" is used, all ports would be updated by this thread. That leaded to traffic loss. Now, when a port is in good condition, the thread doesn't update the MAC and PHY registers for the port.
SDK-108923	1094862	56640_A0 56640_A1 56640_B0	Check for selecting hit bit to clear the L3_DEFIP index is fixed for BCM56640 device.
SDK-108978	1094620	56260_A0 56260_B0 56460_A0 56460_B0	Corrected the check to allow VLAN_PRI to INTERNAL_PRI mapping on SVP for KATANAX devices

SDK-108988	1095388	56860_A0 56860_A1	Counter mode ids were being automatically destroyed when the reference count is 0. Made code changes to avoid deleting mode ids until delete_mode API is called explicitly.
SDK-109000	1095108	56860_A0 56860_A1	Before the code change of this JIRA, a PORTMOD port always returns BCM_PORT_MS_NONE when user calls bcm_port_master_get() to retrieve the SGMII master mode (SGMII Autoneg requires one side of the link partners to be master and the other side to be slave). That is, PORTMOD did not dispatch the request to the actual PHY drivers to retrieve the register values. After the code change, PORTMOD dispatches the request from bcm_port_master_get() to the legacy ext PHY driver if a port is connected to a legacy ext PHY. PORTMOD returns BCM_PORT_MS_SLAVE if a port is pm4x10 or pm4x25 without connected to any ext PHY (the internal PHY can only be in the slave mode). Other than these two cases PORTMOD returns BCM_PORT_MS_NONE to reflect the unknown status without triggering error messages during system init.
SDK-109011	1093521	88660_A0	Egress mirror can be built on COE port and packet recycled with removing COE header. An error appears when mirror delete dynamically that packet will be 1 byte lost in some scenarios. Improvement is done in mirror deletion to fix this error.
SDK-109023	1093399	56860_A1 56865_A1	In previous releases, M_ENABLE bit will be cleared after flex port operation on chip BCM56860. This has been fixed in this release.
SDK-109060	1094198	88670_B0 88680_A0	ILKN 1/3/5 Lane swap configuration: change is not compatible with SDK-6.5.5 and SDK-6.5.6. Compatible with prior versions. When working with ILKN port, it is always recommended to use ILKN lane map SOC property. However, a bug was discovered in 6.5.5 (which was also included in 6.4.11 SP), which caused a change in the ILKN lane map SOC property calculation.

			One outcome of the bug was that the expected SOC property value in 6.5.5 is different than the expected SOC property value in previous versions (only for ILKN1/3/5). This bug is now fixed. With the fix, the expected value for ILKN lane map SOC property is the same as the value in versions prior to 6.5.5.
SDK-109071	1092659	88670_B0 88680_A0	For advanced mode preselection, bcm_field_entry_install was not working for preselectors. Now it is enabled, need to pass the preselector ID together with the stage flag and BCM_FIELD_QUALIFY_PRESEL, for example for ingress stage: bcm_field_entry_install(unit, preselId | BCM_FIELD_QUALIFY_PRESEL | BCM_FIELD_QUALIFY_PRESEL_ADVANCED_MODE_STAGE_INGRESS);
SDK-109078	1095441	56850_A0 56850_A1 56850_A2	In the previous releases, ASF_PORT_CFG can not be dumped by "dump soc". In this release, it can be dumped.
SDK-109119	1093799	88670_B0	In IPv4 MC diagnostic command, "diag pp ipv4_mc" didn't print all the existing entries as expected. Some internal implementation issues were fixed and the diagnostic command works properly now.
SDK-109120	1089702	88670_A0	bcm_cosq_control_set/get fixed to return error status when invalid type is provided with scheduling element gport.
SDK-109460	1096816	88950_a0	When isolating a fabric link, the FE still considers incoming reachability messages as valid. This may cause incorrect reachability reports from the FE. Fixed by modifying isolation sequence. For example: 2x Jericho, 36x FE3200, 1 link between each FE/Jericho After link isolation from FE side, traffic is continuously dropped on FE.

			Solution (implemented internally): 1.Enabling empty reachability cells on FAP side (RcgEnMask=1) 2. New link isolation sequence implemented: a. stop updating unicast tables (setting UCTPUP=0). Maintenance table (RMH) is still being updated. b. Stopping sending reachability on link by masking it in ReachabilityAllowedLinksRegister. c. Wait period of time for: - FE receives empty reachability cell from FAP. - Load balancing takes place. Driver sleeps for 10msec. d. Restore UCTPUP to original value.
SDK-110010	1096336	88670_B0	In previous releases, TX thread was delete by asynchronous mode,in this releases,it is deleted by synchronous mode.
SDK-110015	1080582	88670_A0 88670_B0 88675_A0 88675_B0	Trunk graceful LAG modification for multicast traffic: STANDBY_MC_LB mode could not be enabled with statistics interface enabled (or other non-traffic port such as OLP), and using more than one priority per port. Fixed.
SDK-110184	1097716	88470_A0 88470_B0	MPLS PORT: In the previous release, update operation will override the mac learning property of PWE as true. In this release, it will be fixed, update operation will not change the current mac learning state.
SDK-110185	1097454	56547_A0	Fix overlay selection in cosq_mapping_set
SDK-110228	1097256	88470_A0 88470_B0	OAM: Allow using new trap codes to the OAMP (besides the default one).
SDK-110244	1095348	88750_A0 88750_B0 88950_a0	In case of FE13 device isolation during easy-reload mode isolated link bitmap is being updated, indicating all links have been isolated.

			Unisolation sequence, after exting easy reload mode, will unisolate all links.
SDK-110281	1097060	88670_B0	In previous releases, the FEC ability was set incorrectly when AN enabled; In this release, this issue has been fixed.
SDK-110299		88470_A0 88470_B0	In previous releases, ucode_ports provided the constant interface type XFI when the port type of config.bcm was "XE" and port init speed was 1G. This issue has been fixed in this release.
SDK-110356	1095585	56860_A0 56860_A1	In previous releases, the ports' link status may flap during WB process; In this release, this issue has been fixed.
SDK-110443	1093402	88670_A0 88670_B0	ELI search was disabled as it wasn't working as expected, so when using bcm_switch_control_set with the bcmSwitchHashELISearch option an unavailable error will be returned. A properly working ELI implementation will be added under new LB scheme in version 6.5.8
SDK-110462		88470_A0 88470_B0	In previous releases, Portmod returned invalid value for 10G & XAUI port. This issue has been fixed in this release.
SDK-110502	1099137	56860_A0 56860_A1	In previous releases, when WDRR scheduler mode was set for hsp port, It's not correct for HSP command to shows it as "WERR" . This has been fixed in this release.
SDK-110503	1098068	56540_A0 56540_B0	Compile time definitions for MPLS Entropy was not enabled for Triumph3. This was fixed by adding Triumph3 specific compile time #defines in the SDK code.
SDK-110534		56560_B0 56565_B0	Added soc_feature check to set HG/HG+ on CPU port
SDK-110591	1096909	56850_A0 56850_A1 56850_A2	In previous releases, only 64 child nodes were allowed to attach to their parent node when customer created a LLS tree. This has been fixed in this release by removing this restriction.
SDK-110650	1099231	56960_A0 56960_B0 56960_B1	In the previous release, there was compile error in SAL module when both UNIX and __STRICT_ANSI__ are defined. This has been fixed.
SDK-110772	1090499	56565_A0 56565_B0	Need to Update LLS_PER_PORT_WRR_PKT_MODE_0/1/2 register if port is configured in WRR scheduling mode

SDK-110784	1099255	88375_B0	In previous release, it failed to set speed when existing interface type does not allow such combination. Starting from this release, SDK adopts a default interface type based on the new speed if this speed conflicts with existing interface type.
SDK-110790	1100426	88375_A0 88470_A0	In previous releases, TPID for MinM packet was set with a constant value during system initialization. Now it is configurable by bcm_switch_tpid_*.
SDK-110794	1098378	88375_B0	The outdated bcm_rx_snoop_get API did not return correct ingress probability values. This was fixed. The API that should be used to get information for snoop and mirror profiles is bcm_mirror_destination_get.
SDK-110799	1100326	88675_B0	Which actions/scenarios may trigger the issue: steps as below: 1. load knet modules with parameter "use_rx_skb=1" 2. send packets with the speed of 50p/s and wait to crash What is the impact when the issue occurs: crash with different logs What is the impact of the fix: refine socket buffer operation in knet use_rx_skb mode. 1. from skb to legacy rx, keep the dune headers. (in case of knet mirror from netif to rx) 2. from skb to netif, strip dune headers
SDK-110805	1100530	5340X 5663X 56640	Fix memory lock/unlock mismatch and without freeing allocated buffers before function returns.
SDK-110846		88670_B0	EEDB default entry value is now changed. In previous releases, EEDB use_as_data_entry is set to 1. On some cases, when packet hits EVPN

			program, use_as_data_entry is parsed as outlif. Therefore empty EEDB entry got outlif 1 when parsing by EVPN program. This bug is now fixed. use_as_data_entry is set to 0 by default.
SDK-110875		88660_A0	In previous releases, the lane_select item was set incorrectly for multi-core port when phy diag eyescan was invoked in WarpCoreC; In this release, this issue has been fixed.
SDK-110914	1100685	56460_A0 56460_B0	HW linkscan initialization is added for the ports that are newly created during flex operation for Saber2 devices.
SDK-111009	1100961	56620_B0	Code is added to check whether EGR_SRC_PORT register is available for the device before this register is being accessed.
SDK-111038		88650_A0 88660_A0	In soft error recovery (SER) for exact match memories, recovery procedure may have failed in certain cases, when global LEM counter had to be corrected. An issue in access level was fixed to allow proper recovery procedure.
SDK-111062	1097417	88470_A0 88470_B0	L2: In CPU learning mode, learning events generated by hardware are parsed by SDK first, a proper GPORT is passed to the callback functions to indicate on which port or logical port this MAC address was learned. A problem occurs when a MAC is learned on a forward group gport, in which case SDK may fail to reorgnize the GPORT ID such as vlan port create. With this fix, the MAC learning callback will receive correct GPORT ID and learned entries can be dump correctly.
SDK-111081	1101782	56840_A0	In previous releases, the traffic of WarpCore XAUI was blocked; In this release, this issue has been fixed.
SDK-111117	1096273	88675_A0 88675_B0	Refresh events were disabled in centralized learn mode. Since some events can be handle by DMA and some by MC group, the refresh events should be generated for the MC group. This commit enables the refresh events in centralized learn mode.

SDK-111141		56850_A0 56860_A0	In previous releases, the SW cache of L3_DEFIP table could not be updated correctly during route entry modification on LPM scaling mode. In this release, this has been fixed.
SDK-111243	1100325	88670_A0 88670_B0	Fixed tunnel encapsulation update for BFD sessions. Before the fix, bcm_tunnel_create() returned an error when was called with BCM_TUNNEL_REPLACE flag for BFD sessions.
SDK-111253	1102419	88470_A0 88470_B0	ILKN lane map soc property is now updated to support QAX as well.
SDK-111272	1069717	88670_A0 88670_B0	Update DPP internal sequence to ensure that a port will not get stuck in some scenarios when egress queues are not emptied at a sufficient rate.
SDK-111336	1102247	88670_A0 88670_B0	In the previous release, arad_pp_ihb_flp_program_selection_cam_tbl_get_unsafe() used the wrong macro to get custom_rif_profile_bit. In this release, this issue has been fixed.
SDK-111337	1102303	88470_A0	In previous releases, when changing XFI ports to QSGMII ports through Dynamic Port Provisioning feature, the QSGMII ports could not be attached successfully. This has been fixed in this release.
SDK-111343	1090449	AllChips	Possible stack corruption in port diagnostic application is fixed.
SDK-111378	1101961	88375_B0 88675_B0	In previous releases, the Serdes rate was incorrect when changing QSGMII ports to XFI ports if there was Rx lane swapping in that Port Marco. This has been fixed in this release.
SDK-111420	1093328	88375_A0 88670_B0	When adding a trunk with disabled member to an Egress MC group, the disabled member was also added to the MC group which resulted in mis-routing of traffic to those ports. Fixed.
SDK-111464	1102420	56334 5663X	Fixed Class Type value for bcmFieldQualifySrcClassField qualifier in IFP qualifiers initialization.
SDK-111469	1100635	88670_A0	Drop bit should be set to 1.
SDK-111473		5626X 56271 56270 56272 5645X	Problem: In this case, the SDK programs the bucket count to zero and the refresh count to zero. The problem, though, is that a bucket count of zero indicates that the policer is actually in profile. What this means is that the first packet that hits the policer

		5646X	will cause the packet to take the action associated with the meter being in profile, and then it will decrement the bucket count (causing it to go negative). Since the refresh count is set to zero, the policer will remain out of profile indefinitely, and all subsequent packets will get the out of profile action, which is the desired behavior. Solution: Setting the appropriate bucket count(s) to a negative value if the rate is zero.
SDK-111526	1100530	5340X 5663X 56640	memory free was not handled properly in the error cases. mem unlock also not done for the right memory. Fixed both.
SDK-111527	1100530	5340X 5663X 56640	memory free was not handled properly in the error cases. mem unlock also not done for the right memory. Fixed both.
SDK-111607	1103472	56860_A0 56860_A1	In previous releases, some initial MMU parameters in lossy mode didn't match specification. This has been addressed by updating initial setting.
SDK-111634	1103105	56840_A0	1._stat_err is a internally used static variable, reset it when counter thread is exiting abnormally so that counter thread won't exit right now when customer restarts counter thread and double bit error happens again. 2.Clear the done bit of CMIC_DMA_STAT, the error will be corrected in one period.
SDK-111668	1103922	56547_A0	During entry modify modify, SDK is supposed to modify entries in both partitions when urpf is enabled. Code to update entry in second partition was missing, due to this NHI was not getting update when entry is being replaced. Added code to update entry in second partition when urpf is enabled.
SDK-111715	1098856	56240_B0	Support for default ptpTimescale as per G.725.1 provided.
SDK-111739	1102267	88375_A0 88375_B0	OAM: If user enable Y.1711 soc property, OAM TCAM entry of inner CMF don't set correctly. This issue affects inner CFM packet forwarding .After this fix,

			Y.1711 and inner CFM feature can be coexist, inner CFM packet can be forwarded as default.
SDK-111770		56565_A0	In previous releases, the function _soc_linkscan_port_to_phyaddr of linkscan init didn't get external PHY addr when chip connected to external PHY. This issue has been fixed in this release.
SDK-111772	1101442	88470_A0 88470_B0	encap_id was checked before filling FEC information into hardware, while FEC was allocated successfully in software. if encap_id is invalid, FEC was not filled in hardware. So FEC out of sync between hardware and software tables while API return Error. Now, encap_id will be checked before allocating FEC. if encap_id is invalid, FEC won't be allocated.
SDK-111781		88375_A0 88375_B0	OAM: Terminated CFM packets with 0-2 tags should be forward after PWE termination when PWE MEP is created. The double tag case wasn't supported when MPLSTP MEP is enabled. This only applies when the soc property custom_feature_oam_over_mpls_ignore_mdl is set.
SDK-111814	1104341	56260_A0 56260_B0	LLS display was displaying L0 indices which did not belong to the subport. This was due to incorrect calculation for the start index for subport.
SDK-111830	1104223	88670_A0	Allow to create VSQ-F for recycle and CPU ports. Such VSQ can't be used for flow control generation (only for congestion management).
SDK-111851		88670_A0	OAM: calling bcm_oam_loss_delete() for single ended LM fails to properly update the piggy-back LM/non piggy-back LM status.
SDK-111872	1100895	56640_A0 56640_A1 56640_B0	Scache space is allocated for paritition 1 of mirror module if this space is not allocated during mirror initialization due to being in warm boot mode.
SDK-111873	1097258	56540_A0 56540_B0	During warmboot, ism hash related internal sw database is getting populated in soc_do_init when soc_ism_table_hash_config_get & soc_ism_hash_table_offset_config_get operations are done. But later in soc_misc_init, those value are getting overwritten with default values using soc_ism_table_hash_config &

			soc_ism_hash_offset_config operations. soc_ism_table_hash_config & soc_ism_hash_offset_config should be called only during coldboot and during warmboot. Added check to not overwrite the hash related info during coldboot in soc_misc_init
SDK-112038	1104624	88670_B0	Meter allocation error: Error in the following scenario: bcm_policer_create > bcm_policer_destroy > bcm_policer_create . (same entropy_id). Second create API return an error due to wrong SW profile management. Issue fixed
SDK-112053	1096466	56334_A0	Decrease ref-count for egress object in case of cleanup of existing allocated objects during API cleanup.
SDK-112054	1103311	88670_B0	Fixed: changing existing multicast group may cause DRAM CRC errors: Upon increasing the number of ingress multicast replications in a group, original SDK logic could change the DRAM buffer type consumed by replicated traffic from mini multicast to full multicast. Under traffic, this may occasionally lead to DRAM buffer management unexpected behavior, resulting in a DRAM CRC error condition requiring full reset to recover. In order to eliminate this scenario multicast groups buffer is changed to always use full multicast buffers, regardless of group size.
SDK-112060		56560_B0	Fixed resource leak in xphy_resource_release() which is invoked as part of soc_misc_init()
SDK-112064	1105582	88675_B0	In previous releases, the ports with XLGE2 interface type cannot be configured correctly on BCM82764 external PHY. The speed of those ports were always 100G, which was incorrect. This has been fixed in this release.

SDK-112093	1105352	AllChips	Remove unused files for tr tests.
SDK-112101	1105755	AllChips	In this release, memory corruption in policer module during warmboot is fixed.
SDK-112102	1105036	88470_B0 88680_A0	TR 154 test code is not fit for QAX, JERICHO_PLUS in previous release. Now TR 154 test code has been updated. And the test can be fit for QAX, JERICHO_PLUS.
SDK-112141	1099148	AllChips	Fix changes PP servo state to Acquiring when PTP hybrid mode loses synce signal.
SDK-112159	1103570	88470_A0	When the device init, some interrupts were not cleared. And some interrupts need some special handing. But SDK does not update it. Now these issues have been fixed.
SDK-112391	1106914	88470_A0	OAM: If create egress only mep, the LM counter id can't be updated, This issue affects LM function.After this fix, the LM counter id can be updated correctly.
SDK-112601	1107191	88670_A0 88670_B0	In the previous release, MPLS port replacement operation will override the tpid_profile of PWE as 0. From this release, this issue will be fixed.
SDK-112658	1108119	88470_A0 88470_B0	Fixed issue with the soc property counter_engine_replicated_packets handled wrongly in QAX: 1. CGM MC counting treated this soc property with the opposite meaning. 2. IRPP counting ignored this configuration.
SDK-112898	1100584	88670_A0 88670_B0 88675_A0 88675_B0	When creating an ACL group in external TCAM (KBP), there was a bug when adding an entry to a group which its key is greater than 80 bits, and not byte aligned. The issue solved in FP, to copy the source buffer to the KBP DB in bit resolution, instead of byte resolution.

Section 10: Unresolved Issues for 6.5.7

The following open Urgent priority issues remain unresolved in version 6.5.7 of the SDK.

Number	*CSP*	*Chips*	*Errata For 6.5.7*
PHY-2019	976013	56340_A0	SDK build fails when compiled with bmacsec for iproc devices
SDK-91219	1014281	88670_A0 88670_B0 88950_A0	Using bcm_port_phy_control get for Falcon RX settings leaves uc stopped
SDK-108447	1092047	88375_A0 88375_B0 88670_A0 88670_B0 88950_A0	SOC property custom_feature_ams_pll_override not working in SDK 6.5.5
SDK-108877	1092910	88470_A0	ILKN link down with DFE enabled
SDK-111509	832200	56634_A0 56634_B0 56636_A0 56636_B0 56638_A0 56638_B0 56639_A0	Missing rx_reasons for L3VPN packets copy to CPU on SDK 6.3.3
SDK-112610	1107233	56960_A0 56960_B0 56960_B1	Creating VFP (FP Stage Lookup) in multiple pipelines fail. Worked in SDK 6.4.7 but fails in 6.5.3
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-6.5.7-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 general 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 for full details.

	SDK-6.5.7	SDK-6.5.6	SDK-6.5.5	SDK-6.5.4	SDK-6.5.3	SDK-6.4.11
4.3.3	BCM56760 BCM56565	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

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
6.5.7	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	6.5.7 Firmware Versions
BCM84861	00.00.10	00.00.10	01.00.00	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	01.00.00
BCM84868	00.00.10	00.00.10	01.00.00	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	01.03.02
BCM84856	01.02.10	01.02.10	01.03.02	01.03.02	01.03.02	01.03.02
BCM84744	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0130 (B0/C0)	0x0130 (B0/C0)
BCM84757	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)
BCM84328	R027	R029	R029	R029	R029	R029
BCM82322/28 B1	0xF	0xF	0xF	0xF	0xF	0xF
BCM82780	0x23	0x23	0x23	0x23	0x23	0x28
BCM82752	0x23	0x23	0x23	0x23	0x23	0x28
BCM82758	0x23	0x23	0x23	0x23	0x23	0x28
BCM82764	D011	D013	D017	D017	D017	D019
BCM82790	D011	D013	D017	D017	D017	D019
BCM82792	D011	D013	D017	D017	D017	D019

BCM82796	D011	D013	D017	D017	D017	D019
BCM82381	D011	D013	D013	D013	D013	D015
BCM82209	D011	D013	D013	D013	D013	D015
BCM82073	D011	D013	D013	D013	D013	D015
BCM82864	N/A	N/A	D00C	D00C	D00C	D00E
BCM82332	N/A	N/A	D006	D008	D008	D009
Eagle	D10F_05	D10F_OD	D10F_0D	D10F_0D	D10F_13	D10F_13
Falcon	D10A_07	D10B_07	D10B_0C	D10B_0E	D10B_13	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.

Section 14: Feedback

Broadcom engineering values our customers' feedback on our releases and release notes documentation. In order to improve the material and provide customers with the most pertinent information, please follow this survey link to provide your feedback on the current release. Alternatively, you may email ron.meadows@broadcom.com with any free form feedback as well. We will consider implementing your suggestions for improvement in future versions of our releases and documentation.