Software Development Kit
Release Notes
SDK 6.4.2

September 30, 2014

Broadcom
Network Switching

Section 1: About This Document

These are the Release Notes for the Broadcom Network Switching Software Development Kit Release 6.4.2.

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.

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:

Table 1: Product Documentation

Document	*Description*
56XX-PG642-R	BCM API Reference Guide. This manual describes the theory of operations of the API and all existing BCM APIs for this release.
56XX-PG707-R	Stacking Software Guide This guide describes how to use the discovery and stacking applications provided in this release.
56XX-PG818-R	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.

Section 3: New in this Release

This section describes feature and device support that is introduced in this release.

SUMMARY OF NEW FEATURES

BRING-UP SUPPORT - BCM56960

BCM56960 family is industry's First High-Density 25/100 Gigabit Ethernet Switch for Cloud-Scale Networks. The switch offers 32 100GbE, 128 25GbE, ports. This release contains Early Access bring up support for BCM56960 A0

This release supports the following device configurations.

- BCM56960 A0: 32x100GbE, 128x10GbE

Status of support:

- Basic bring up of SDK on hardware is completed.
- Diagnostic tests: All of the TR tests run with loopback are passing.
  - Register/Memory access test
  - Table DMA test
  - Linkscan MDIO test
  - Traffic Test with MAC, internal and external PHY loopback
  - Counter/MIB test
  - L2/L3 Table operation test
  - VLAN table operation test
  - COSQ test
- State of PHY Microcode:
  - The PHY microcode delivered in this release is only meant for bringup purposes. It is not guaranteed to withstand stress testing (such as overnight runs on every port). It is partially regressed to be reasonably stable for typical process and temperatures. It's adaptation may vary slightly between ports depending on channel variations.

While the device and legacy feature bringup are still in progress, the summary of the current base test status as of this release are as follows:

Table 2: Status of the supported features (BCM APIs)

Features	Status	Unit Test	Regression Results on Hardware (% Fail)	Note
Legacy				
alpm legacy	Not supported			Bringup on Hardware in progress
Cosq	Preview	HW	5.5%	
Cut Through Mode	Not supported			Bringup on Hardware in progress
extender	Preview	HW	0.0%	
failover	Preview	HW	0.0%	
fcoe	Preview	HW	0.0%	
field	Not fully supported			VFP and EFP are supported. IFP legacy support (no compression) is not yet integrated will be available in an EA release in late October
Flex stat	Preview	BCMSIM	NA	Implemented with exception of IFP Bringup on HW in progress
Higig Proxy	Preview	HW	0.0%	
IPMC Legacy	Preview	HW	3.2%	

Table 2: Status of the supported features (BCM APIs)

Features	*Status*	*Unit Test*	*Regression Results on Hardware (% Fail)*	*Note*
Flexport	Not supported			Coded but no testing on HW yet
L2	Beta	HW	0.4%	
L2GRE	Preview	HW	0.0%	
L3	Beta	HW	1.7%	
link	Preview	HW	0.0%	Bringup on HW in progress
MIM	Preview	HW	0.0%	
mirror	Preview	HW	4.1%	Bringup on HW in progress
misc	Preview	HW	0.0%	
MPLS	Preview	HW	1.6%	
Multicast	Beta	HW	0.0%	
NAT	Preview	HW	Inc in L3	
NIV Legacy	Preview	HW	4.3%	Bringup on HW in progress
Packet Rx/Tx	Preview	HW	0.0%	
PIMDIR	Preview	HW	Inc in L3	
port	Preview	HW	7.3%	See SerDes bringup details below
proxy	Preview	HW	39.7%	Dependent on IFP completion
QoS	Preview	HW	0.0%	
Oversub	Not supported yet			
rate	Preview	HW	0.0%	
Rtag 7 - flex hashing	Not supported yet			
SER Correction	Not supported yet			Mostly coded but not yet tested on HW
Stack	Preview	HW	0.0%	
STAT	Preview	HW	3.0%	
STG	Preview	HW	0.0%	
Switch Controls	Preview	HW	3.2%	
Time	Preview	HW	0.0%	
TRILL	Preview	HW	2.5%	
Trunk	Preview	HW	4.0%	Bringup on HW in progress
Tunnel	Preview	HW	6.2%	Bringup on HW in progress
Virtual/subport	Not supported yet			
VPLAG	Preview	HW	Inc in Trunk	
VLAN	Preview	HW	3.4%	Failures largely related to field functions not yet complete
VXLAN	Beta	HW	0.0%	
warmboot	Not supported yet			
wlan	Preview	HW	0.0%	
New Features				
Visibility - Packet Trace	Not supported yet			Data abstraction not yet complete - feature disabled
Counter Ejection	Preview	HW	NA	Test development in progress
IPMC Enhancements	Not supported yet			
VP Switch Routing	Not supported yet			

Table 2: Status of the supported features (BCM APIs)

Features	Status	Unit Test	Regression Results on Hardware (% Fail)	Note
MPLS LSR	Not supported yet			
Latency Bypass	Not supported yet			
Hierarchical ECMP	Not supported yet			
Hash Spray	Not supported yet			
Hashing - Load Balancing	Not supported yet			
Sflow and general rate enhancements	Not supported yet			
Visibility - Aggregate and PFC Monitors	Not supported yet			
New NIV Features	Not supported yet			
DTCP	Not supported yet			
AVS	Not supported yet			
IFP Manual Compression	Not supported yet			
IFP Auto Compression	Not supported yet			
SerDes Interfaces				
4x10				
10GBASE-XFI, Forced Speed	Passed (Gloop, PRBS (port to port, Traffic(port to port)))			Traffic is CPU generated
4x25				
40G-MLD, Forced Speed	Passed (Gloop, PRBS (port to port, Traffic(port to port)))			Traffic is CPU generated
100G-MLD, Forced Speed	Passed (Gloop, PRBS (port to port, Traffic(port to port)))			Traffic is CPU generated
Link Training	Not supported yet			
Autonegotiation	Not supported yet			
Eyescan HighBER	Not supported yet			
Eyescan LowBER	Not supported yet			
Other Speeds	Not supported yet			
HG	Not supported yet			

BCM88670 (JERICHO) PREVIEW RELEASE

The Broadcom(R) BCM88X7X product line is the sixth generation of the Dune product line devices. Together with the BCM88950 fabric element (FE) device, it is used to build a variety of network switch solutions, enabling switching platforms of up to 12,000 100G Ethernet ports. The BCM88X7X device processes up to 720 Gbps traffic at wire-speed with various port rate combinations, supporting up to six 100G full- duplex ports at Layer 2 through Layer 4, with integrated deep-buffer traffic management capabilities and a fabric interface.

The following features were implemented in this release on top of previous version:

- Multicast, backward-compatible formats
- Inbound and outbound mirroring

- Trunk (LAG)
- Interrupts handling infrastructure

BCM88950 (FE3200) PRE SILICON RELEASE

Broadcom BCM88950 is the fourth generation in the Dune product line of Fabric Element (FE) devices - following BCM88750 (FE1600). This release is pre-silicon, and `should not` be used for BCM88950 bring-up. BCM88950 bring-up release, 6.4.3-EA1, will be released shortly after 6.4.2, and will include fixes following BCM88950 silicon validation, but will not be validated for other devices.

The following BCM88950 exclusive features were implemented in 6.4.2 compared to the 6.4.2 EA2 version:

- Inband CPU packets
- Isolation and graceful shutdown
- Synchronous Ethernet (SyncE)

GREYHOUND

BCM534XX family with switch and embedded processor SOC chips offers industry-leading integration and performance in a small footprint. Up to 16 10GbE ports are supported with KX, KR, XAUI, RXAUI, and SGMII modes. In addition to the device SKUs supported in previous release, this release contains Early Access support for some more devices in external and internal host modes that have been brought up in house.

The devices that have been in bring-up complete state with respective port configurations are as listed: (x denotes as 0/1)

- BCM534x2: 8 x 1G/2.5G/5G/10G
- BCM534x5: 16 x 1G/2.5G/5G/10G

The devices of Elkhound (BCM5345x) and Bloodhound (BCM5342x) are also in preview state, which actual device IDs are listed in the section of Preview Switch Devices.

THINGS TO NOTE

This section lists items that require special attention.

BCMX API DEPRECATION

BCMX APIs have not been enhanced or supported for newer devices since SDK-5.10.2. Legacy BCMX APIs, supported in SDK-5.10.2 will be deprecated starting with SDK-6.3.5 release. Customers are encouraged to transition from BCMX APIs to their equivalent BCM APIs. Please contact Broadcom application support for any help in the transition.

BCM8483X PHY FIRMWARE

Status of F/W version 1.69 released with SDK is preview. Check support.broadcom.com for latest available validated firmware for the BCM8483X family devices. Consult F/W release notes for known issues.

BCM8484X PHY FIRMWARE

Status of F/W version 1.07.11 released with SDK is preview. Check support.broadcom.com for latest available validated firmware for the BCM8484X family devices. Consult F/W release notes for known issues.

SPN_PHY_PORT_PRIMARY_AND_OFFSET

Setting of the config property `spn_PHY_PORT_PRIMARY_AND_OFFSET` is absolutely required for the following PHYs. BCM54880E BCM54680E BCM54682E BCM54685E BCM54640E BCM542XX

BCM56850 HANDLING OF MODULE_64PORTS

Setting `module_64ports=1` config variable indicates that one module ID covers 64 ports. Please, note that while on the earlier devices this effectively meant choosing a single-modid mode for the device due to the fact that the total number of ports was less than 64, on devices that have more than 64 ports, such as BCM56850 (what about 56840?) it means choosing dual-modid mode instead. Please, always use `bcm_stk_modid_count()` to get the actual number of modids required by a given device. Also, note that the API `bcm_port_gport_get()` is the only correct way to translate the physical port number into a `MODPORT_GPORT` and it works correctly regardless of the number of modids assigned to the device

OCCASIONAL STACK ATTACH FAILURES

There may be occasional stack attach failures due to the stack master attaching a slave device before slave programming is complete. Use the stk.soc config variable stktopomasterdelay to increase the length of time the stack master will wait before attaching a slave.

UNBALANCED MUTEX WARNING

A potential issue with unbalanced mutexes has been uncovered in previous releases of SDK and special code has been added to automatically detect that condition. While we believe that we've identified all these issues in our regression testing, there is a slight probability that you can see the following message on the console:

WARNING: Mutex `"mutex_name"` has not been unlocked before being destroyed.

Current owner is `"thread_name".`

Should you see such a warning, please, copy it verbatim and contact Broadcom Support.

IP ROUTE LOOKUP

If a packet destination IP lookup falls in between 2 route prefixes having a common prefix part and differing lengths, then the lower route prefix, which is the correct match, may not always be returned correctly.

BCM FIELD QUALIFIER TUNNEL TYPE

The enumeration type `bcm_field_TunnelType_t` has changed its values between SDK-6.2.x, SDK-6.3.0 and SDK-6.3.1. The implication is that legacy field qualifier support for `bcm_field_qualify_TunnelType` in XGS devices running SDK-6.3.1 and later has been broken. Features such as RPC between systems running SDK-6.2.x or SDK-6.3.0 on one and SDK-6.3.1 (or later) on another will not work properly for the BCM field qualifier tunnel type. This issue is being addressed in SDK-6.3.5, SDK-6.4.0, and later releases.

WARMBOOT: VALIDATED WARMBOOT UPGRADES.

Following warmboot upgrades have been validated in this release.

Table 3: Validated Warmboot upgrades

Software upgrade	*Supported*
6.4.1 to 6.4.2	Yes
6.3.9 to 6.4.2	Yes

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

- Trident2
- Trident+
- Triumph3
- Katana2
- Helix4
- Hurrican2
- Katana
- Raven
- Saber
- Greyhound

NEW SPINLOCK APIS

In the SDK release v6.3.8/SDK 6.4.1 a new lock mechanism -- spinlock was introduced to satisfy some requirements for protecting small critical sections more efficiently. The spinlock mechanism is applicable to the scenario in which the critical section to be protected only contains simple operations, such as inserting or removing nodes from a linked list, increasing or decreasing shared data. The data structure and interfaces of the spinlock are defined at SAL layer in SDK as follows,

```
typedef struct sal_spinlock_s {
    char spinlock_opaque_type;
} *sal_spinlock_t;

sal_spinlock_t sal_spinlock_create(char *desc);
int sal_spinlock_destroy(sal_spinlock_t lock);
int sal_spinlock_lock(sal_spinlock_t lock);
```

```
   int sal_spinlock_unlock(sal_spinlock_t lock);
```

These primitives have been implemented for vxWorks and Linux. The changes are available in src/sal/core/linux/sync.c for Linux kernel mode, src/sal/core/unix/sync.c for Linux user mode and src/sal/core/vxworks/sync.c for vxWorks. Customers who use different OSes will need to make similar implementation in their OS specific SAL layer source files. If additional information is needed, please refer to the field alert document "Spinlock Application Note" or contact your Field Support staff.

NEW DEVICES AND SYSTEMS

For any given SDK release, support for certain devices may be provided in Preview or Supported status. Devices in preview status are provided to allow early integration of the customer's application with the SDK APIs that support that device. This software has not been tested on the physical target device and should not be expected to fully function.

Devices in "Supported" status 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.

Table 4: Preview Switch Devices

Family	Devices	Description
BCM53400	BCM53454 A0	20 x 1G/2.5G + 4 x 1G/2.5G/5G/10G
	BCM53455 A0	20 x 1G/2.5G + 4 x 1G/2.5G/5G/10G with embedded A9
	BCM53456 A0	2 x QSGMII + 16 x 1G/2.5G + 4 x 1G/2.5G/5G/10G (option3)
	BCM53457 A0	2 x QSGMII + 16 x 1G/2.5G + 4 x 1G/2.5G/5G/10G with embedded A9 (option3)
	BCM53422 A0	8 x 1G + 2 x 1G/2.5G/5G/10G
	BCM53424 A0	2 x QSGMII + 16 x 1G + 4 x 1G/2.5G/5G/10G (option3)
	BCM53426 A0	20 x 1G + 4 x 1G/2.5G/5G/10G
BCM56960	BCM56960 A0	32x100 GbE/64x40GbE/128x10 GbE Multilayer Switch
BCM56450	BCM55450 B0	KT2 Access-8 FX + 2 F-HG
BCM56450	BCM56452 B0	24xGE + 4xF.XAUI
BCM56450	BCM56454 B0	8xGE + 2 x F.XAUI
BCM56455	BCM55455 B0	KT2 Access - 8 FX + 2 F-HG
BCM56456	BCM56457 B0	24xGE + 4xF.XAUI
BCM56456	BCM56458 B0	8xGE + 2xF.XAUI
BCM56640	BCM56044	Ranger+ SKU - 100G + 3xF.HG[42] + 1GE
BCM56640	BCM56543	Apollo2+ SKU . Bringup -100GE/3xF.HG[42] + F.HG[127] + 1GE 415MHz
BCM56640	BCM56545K	Triumph 3 SKU - Bringup - 48-port GE switch + 4x10GE + 4xHG[42] / 40GE
BCM56640	BCM56546K	Triumph 3 SKU - Bringup - 28-port GE switch + 4x10GE + 4xHG[42] / 40GE
BCM56240	BCM56240	Saber SKU - Bringup - 2x (10GbE/4x 1GbE/4x 2.5GbE) + 2x 10GbE/12GbE/13GbE, IEEE 1588 enable
BCM56240	BCM56240	Saber SKU - Bringup - 10x 1GbE/2.5GbE, IEEE 1588 enabled
BCM56340	BCM56345	Twister . Bringup - 12xF.QSGMII + Flex[4x10] + 1GE
BCM56340	BCM56346	Twister . Bringup - 7xF.QSGMII + Flex[4x10] + 1GE
BCM56450	BCM56450 B1	24-port GbE Multilayer Switch with 4-port 10 GbE uplinks, stacking, integrated CPU and Traffic Manager

Table 5: PHYs

Device	Driver Family	Description
BCM54618_A0	54616	Single-Chip 10/100/1000BASE-T Gigabit Ethernet Transceiver (IEEE 1588 features are not supported by SDK driver)
BCM82328_B0	82328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY.
BCM54220SE	54210	Dual Copper/Fiber Gigabit Ethernet Transceiver, EEE, SyncE supported (IEEE 1588 not yet supported by SDK driver)
BCM82322_B0	82328	12port Gallardo28 supporting 12x10G, 3x40G, 1x100G.

Table 6: Preview PHYS

Device	Driver Family	Description
BCM84858_A0	84858	Quad 10GBASE-T Transceiver. Firmware version 00.03.00
BCM82328_B0	82328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version D
BCM82322_B0	82328	12port Gallardo28 supporting 12x10G, 3x40G, 1x100G. Firmware version D
BCM54210SE_B0	54210	Single Copper/Fiber Gigabit Ethernet Transceiver. 1588 not yet supported
BCM54220SE_B0	54220	Dual Copper/Fiber Gigabit Ethernet Transceiver. 1588 not yet supported
BCM82072	82072	CAUI4-TO-KR4 NRZ BACKPLANE PHY/OCTAL 25G BACKPLANE RETIMER. (Bringup)
BCM54295SE_B0	54295	Octal Copper/Quad Fiber Gigabit Ethernet Transceiver (Bringup) EEE , SyncE and 1588 not yet supported
BCM54296SE_B0	54296	Quad Copper/Fiber Gigabit Ethernet Transceiver (Bringup) EEE , SyncE and 1588 not yet supported

Table 7: Preview CPU Subsystems

CPU Subsystem	Description
BCM9XLP2_XMC_A1	XMC with Broadcom XLP II 200 series multicore processor (MIPS64 Release-II ISA-compliant) with eight NXCPU. processing units, each operating at up to 2.0 GHz

SUMMARY OF BCM CHANGES

This section summarizes BCM API changes in this release. Complete documentation is available in the Network Switching Software Programmer's Guide 56XX-PG642-R. (See section 2 earlier in this document for availability).

BCM FUNDAMENTALS

BCM_MODID_ALL is newly defined and introduced in this release.

New data types bcm_core_t has been introduced in this release.

```
        typedef int32 bcm_core_t;
        typedef int bcm_policer_pool_t;
```

New GPORT macros have been added for subscriber group.

Table 8: New Macros for Analyzing/Constructing GPORTs

GPORT Macros	*Macro Description*
BCM_GPORT_UCAST_SUBSCRIBER_QUEUE_GROUP_SYSQID_SET(g, port_id, qid)	Sets subscriber group type, port id and queue value
BCM_GPORT_UCAST_SUBSCRIBER_QUEUE_GROUP_SYSPORTID_GET(g)	Get portid from the subscriber gport
BCM_GPORT_LOCAL_FABRIC_SET	Set gport from fabric link
BCM_GPORT_LOCAL_FABRIC_GET	Get local fabric link from gport
BCM_GPORT_IS_LOCAL_FABRIC	Returns 1 if gport is local fabric
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_UCAST_SET	Set gport as fabric unicast FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_MCAST_SET	Set gport as fabric multicast FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_TDM_SET	Set gport as fabric TDM FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_LOCAL_UCAST_SET	Set gport as fabric Local unicast FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_LOCAL_MCAST_SET	Set gport as fabric Local multicast FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_LOCAL_TDM_SET	Set gport as fabric Local TDM bypass FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_UCAST_CORE_GET	Get gport from fabric unicast FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_MCAST_CORE_GET	Get gport from fabric multicast FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_TDM_CORE_GET	Get gport from fabric TDM FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_LOCAL_UCAST_CORE_GET	Get gport from fabric Local unicast FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_LOCAL_MCAST_CORE_GET	Get gport from fabric Local multicast FIFO
BCM_COSQ_GPORT_FABRIC_RX_QUEUE_LOCAL_TDM_CORE_GET	Get gport from fabric Local TDM bypass FIFO
BCM_COSQ_GPORT_IS_FABRIC_RX_QUEUE_UCAST	Returns 1 if gport is fabric unicast FIFO

Table 8: New Macros for Analyzing/Constructing GPORTs

GPORT Macros	Macro Description
BCM_COSQ_GPORT_IS_FABRIC_RX_QUEUE_MCAST	Returns 1 if gport is fabric multicast FIFO
BCM_COSQ_GPORT_IS_FABRIC_RX_QUEUE_TDM	Returns 1 if gport is fabric TDM FIFO
BCM_COSQ_GPORT_IS_FABRIC_RX_QUEUE_LOCAL_UCAST	Returns 1 if gport is fabric Local unicast FIFO
BCM_COSQ_GPORT_IS_FABRIC_RX_QUEUE_LOCAL_MCAST	Returns 1 if gport is fabric Local multicast FIFO
BCM_COSQ_GPORT_IS_FABRIC_RX_QUEUE_LOCAL_TDM	Returns 1 if gport is fabric Local TDM bypass FIFO
BCM_COSQ_GPORT_CORE_SET(g, core)	Sets core type and value
BCM_COSQ_GPORT_IS_CORE	Returns 1 if core type

BCM MODULES

A new module BCM_MODULE_SAT has been added in this release.

The BCM APIs are classified into API groups called 'modules'. The following table lists all the BCM modules and their string equivalents.

Table 9: BCM Module IDs

BCM Module ID	String Equivalent
BCM_MODULE_PORT	port
BCM_MODULE_L2	l2
BCM_MODULE_VLAN	vlan
BCM_MODULE_TRUNK	trunk
BCM_MODULE_COSQ	cosq
BCM_MODULE_MCAST	mcast
BCM_MODULE_LINKSCAN	linkscan
BCM_MODULE_STAT	stat
BCM_MODULE_VIRTUAL	virtual
BCM_MODULE_COMMON	common
BCM_MODULE_MIRROR	mirror
BCM_MODULE_L3	l3
BCM_MODULE_STACK	stack
BCM_MODULE_IPMC	ipmc
BCM_MODULE_STG	stg
BCM_MODULE_TX	tx
BCM_MODULE_L2GRE	l2gre
BCM_MODULE_AUTH	auth
BCM_MODULE_RX	rx
BCM_MODULE_FIELD	field
BCM_MODULE_TIME	time
BCM_MODULE_SUBPORT	subport
BCM_MODULE_MPLS	mpls
BCM_MODULE_FABRIC	fabric
BCM_MODULE_MIM	mim

Table 9: BCM Module IDs

BCM Module ID	String Equivalent
BCM_MODULE_POLICER	policer
BCM_MODULE_OAM	oam
BCM_MODULE_FAILOVER	failover
BCM_MODULE_VSWITCH	vswitch
BCM_MODULE_WLAN	wlan
BCM_MODULE_QOS	qos
BCM_MODULE_MULTICAST	multicast
BCM_MODULE_TRILL	trill
BCM_MODULE_IPFIX	ipfix
BCM_MODULE_NIV	niv
BCM_MODULE_CES	ces
BCM_MODULE_PTP	ptp
BCM_MODULE_BFD	bfd
BCM_MODULE_REGEX	regex
BCM_MODULE_VXLAN	vxlan
BCM_MODULE_EXTENDER	extender
BCM_MODULE_FCOE	fcoe
BCM_MODULE_UDF	udf
BCM_MODULE_SAT	sat

BIDIRECTIONAL FORWARDING DETECTION

New fields of data structure `bcm_bfd_endpoint_info_t`, which contains information on both the local and remote ends of the BFD session, have been added as well as the new flags for BFD Endpoint Info Structure Flag Definitions.

Table 10: BFD Endpoint Info Structure Flag Definitions

Flag	Description
BCM_BFD_ENDPOINT_HW_ACCELERATION_SET	Configure only HW accelerator properties.
BCM_BFD_ECHO	Create BFD echo session.

```
typedef struct bcm_bfd_endpoint_info_s {
    ...
    /* For local endpoints, this is the BFD transmission period in ms. */
    int bfd_period;
    ...
    /* MPLS-TP CC/CV TLV and Remote MEP ID. */
    uint8 remote_mep_id[BCM_BFD_ENDPOINT_MAX_MEP_ID_LENGTH];
    /* Length of MPLS-TP CC/CV TLV and REMOTE MEP-ID. */
    uint8 remote_mep_id_length;
    /* MPLS-TP CC/CV TLV and Mis connectivity MEP ID. */
    uint8 mis_conn_mep_id[BCM_BFD_ENDPOINT_MAX_MEP_ID_LENGTH];
    /* Length of MPLS-TP CC/CV TLV and Mis connectivity MEP-ID. */
    uint8 mis_conn_mep_id_length;
} bcm_bfd_endpoint_info_t;
```

The data types for BFD Statistics structure have been extended from `uint32` to `uint64`.

```
    typedef struct bcm_bfd_endpoint_stat_s {
        uint64 packets_in;
        uint64 packets_out;
        uint64 packets_drop;
        uint64 packets_auth_drop;
    } bcm_bfd_endpoint_stat_t;
```

Some other miscellaneous BFD defines are added.

Table 11: Miscellaneous BFD defines

Define Name	Description
BCM_BFD_ENDPOINT_DEFAULT1	BFD default enbdpoint for OAM trap profile 1
BCM_BFD_ENDPOINT_DEFAULT2	BFD default enbdpoint for OAM trap profile 2
BCM_BFD_ENDPOINT_DEFAULT3	BFD default enbdpoint for OAM trap profile 3

Below new APIs are added for this release.

bcm_bfd_tx_start

Start the Transmission of BFD PDUs.

Syntax

```
#include <bcm/bfd.h>
int
bcm_bfd_tx_start(
    int unit);
```

Parameters

unit	(IN) Unit number.

Description

Start the Transmission of BFD PDUs related to all endpoints.

Returns

bcm_bfd_tx_stop

Stop the Transmission of BFD PDUs.

Syntax

```
#include <bcm/bfd.h>
int
bcm_bfd_tx_stop(
    int unit);
```

Parameters

`unit`	(IN) Unit number.

Description

Stop the Transmission of BFD PDUs related to all endpoints.

Returns

REGEX API

A new enumerated flag bcmStatRegexSigMatchLast to denote End of SigMatch part of `bcm_regex_stat_t` list has been added in this release.

A new field `flow_end_reason` has been added in `bcm_regex_report_t` data structure with its supported reasons.

```
/* Regex report structure. */
typedef struct bcm_regex_report_s {
   ...
   bcm_regex_flow_end_reason_t flow_end_reason; /* FT flow end reason */
   ...
} bcm_regex_report_t;
```

Table 12: Flow Tracker's Flow End Reasons

heading Flow End Reason	***heading Description***
bcmRegexFlowEndReasonFirst	Beginning of Flow End Reason
bcmRegexFlowEndReasonTcpAgeout	Flow session ended because of age out
bcmRegexFlowEndReasonTcpFINr	Flow session ended because TCP FIN received in reverse direction
bcmRegexFlowEndReasonTcpFINf	Flow session ended because TCP FIN received in forward direction
bcmRegexFlowEndReasonTcpReset	Flow session ended because TCP RESET received in either direction

CLASS OF SERVICE QUEUE CONFIGURATION

A new flag for bandwidth response tuning has been added for `bcm_cosq_port_bandwidth_get/set()` APIs.

Table 13: Port Bandwidth Flags

Flag	Description
BCM_COSQ_BW_NOT_COMMIT	Set rate to SW DB only, commit to HW is done when calling the API with teh same GPort type and without NOT_COMMIT flag

A new COSQ control type bcmCosqControlSingleCalendarMode has been added in this release.

Table 14: CoSQ Control Type Values

Value	Description	Arg value
bcmCosqControlSingleCalendarMode	Set single calendar mode for an interface.	value: 0-disable, 1-enable

Two fields has been added into the data structure for BCM COSQ DELAY TOLERANCE Configuration, where the `BCM_COSQ_DELAY_TOLERANCE_SLOW_LEVELS` is defined to 7.

Table 15: BCM COSQ DELAY TOLERANCE Configuration

Parameter / Description
slow_level_thresh_up - Seven slow levels thresholds upwards.
slow_level_thresh_down - Seven slow levels thresholds downwards.

Table 16: BCM XCore CoSQ GPORT Modes

Mode	Description
BCM_COSQ_DELAY_TOLERANCE_[08-23]	Flexible delay tolerance level.
BCM_COSQ_DELAY_TOLERANCE_SLOW_LEVELS	Number of slow threshold levels.

Below two new flags supported by the flags field of `bcm_cosq_delay_tolerance_t` have been added.

Table 17: flags supported by the flags field of bcm_cosq_delay_tolerance_t

Flag	Description
BCM_COSQ_DELAY_TOLERANCE_HIGH_Q_PRIORITY	Specifies that the profile is for high priority queue
BCM_COSQ_DELAY_TOLERANCE_TOLERANCE_OCB_ONLY	Specifies that the profile is for OCB-only queues

One new field `core_id` has been added to `bcm_cosq_vsq_info_t` data type for VSQ info setting.

Table 18: VSQ info settings

Parameter	Description
core_id	The core ID of the VSQ

The flag BCM_COSQ_DISCARD_VSQ_OCB_ONLY has been added as the latest addition of discard options for bcm_cosq_gport_discard_get/set() APIs. BCM_COSQ_GPORT_SIZE_VSQ_OCB_ONLY is also defined for the flag of bcm_cosq_gport_color_size_set().

Table 19: gport Discard Color Flags

Flag	Description
BCM_COSQ_DISCARD_VSQ_OCB_ONLY	Configure the OCB-only VSQ, else configure the Dram-mix VSQ.

Several VSQ GPORT handle MACROs have been added.

Table 20: VSQ GPORT handle MACROs: used to handle the different VSQ types, and their static attributes.

BCM_COSQ_GPORT_VSQ_GL_SET(_gport, _core_id)	Corresponding to gports created with BCM_COSQ_VSQ_GL flags
BCM_COSQ_GPORT_IS_VSQ_GL(_gport)	Corresponding to gports created with BCM_COSQ_VSQ_GL flags
BCM_COSQ_GPORT_VSQ_CT_SET(_gport, _core_id, _category)	Corresponding to gports created with BCM_COSQ_VSQ_CT flags
BCM_COSQ_GPORT_IS_VSQ_CT(_gport)	Corresponding to gports created with BCM_COSQ_VSQ_CT flags
BCM_COSQ_GPORT_VSQ_CTTC_SET(_gport, _core_id, _category, _traffic_class)	Corresponding to gports created with BCM_COSQ_VSQ_CTTC flags
BCM_COSQ_GPORT_IS_VSQ_CTTC(_gport)	Corresponding to gports created with BCM_COSQ_VSQ_CTTC flags
BCM_COSQ_GPORT_VSQ_CTCC_SET(_gport, _core_id, _category, _connection_class)	Corresponding to gports created with BCM_COSQ_VSQ_CTCC flags
BCM_COSQ_GPORT_IS_VSQ_CTCC(_gport)	Corresponding to gports created with BCM_COSQ_VSQ_CTCC flags
BCM_COSQ_GPORT_VSQ_CT_GET(_gport)	Get connection class
BCM_COSQ_GPORT_VSQ_CORE_ID_GET(_gport)	Get core id
BCM_COSQ_GPORT_VSQ_TC_GET(_gport)	Get traffic class
BCM_COSQ_GPORT_VSQ_PP_SET(_gport, _statistics_tag)	Corresponding to gports created with BCM_COSQ_VSQ_PP flags
BCM_COSQ_GPORT_VSQ_PP_GET(_gport)	Corresponding to gports created with BCM_COSQ_VSQ_PP flags
BCM_COSQ_GPORT_IS_VSQ_PP(_gport)	Corresponding to gports created with BCM_COSQ_VSQ_PP flags
BCM_COSQ_GPORT_VSQ_SRC_PORT_SET(_gport, _src_port_vsq_id)	Corresponding to src port gports created with API bcm_cosq_src_vsqs_gport_add
BCM_COSQ_GPORT_VSQ_SRC_PORT_GET(_gport)	Corresponding to src port gports created with API bcm_cosq_src_vsqs_gport_add
BCM_COSQ_GPORT_IS_VSQ_SRC_PORT(_gport)	Corresponding to src port gports created with API bcm_cosq_src_vsqs_gport_add

Table 20: VSQ GPORT handle MACROs: used to handle the different VSQ types, and their static attributes.

BCM_COSQ_GPORT_VSQ_GL_SET(_gport, _core_id)	***Corresponding to gports created with BCM_COSQ_VSQ_GL flags***
`BCM_COSQ_GPORT_VSQ_PG_SET(_gport, _pg_vsq_id)`	Corresponding to pg port gports created with API `bcm_cosq_src_vsqs_gport_add`
`BCM_COSQ_GPORT_VSQ_PG_GET(_gport)`	Corresponding to pg port gports created with API `bcm_cosq_src_vsqs_gport_add`
`BCM_COSQ_GPORT_IS_VSQ_PG(_gport)`	Corresponding to pg port gports created with API `bcm_cosq_src_vsqs_gport_add`

New threshold flags and types for the configuration of flow control and admission control have been added for data type `bcm_cosq_threshold_t` for `bcm_cosq_gport_threshold_set/get()` APIs.

Table 21: Threshold flags

BCM_COSQ_THRESHOLD_PER_DP	***dp field meaningful***
`BCM_COSQ_THRESHOLD_OCB`	On-Chip buffer
`BCM_COSQ_THRESHOLD_ETH_PORT_LLFC`	Set LLFC threshold value for ethernet port
`BCM_COSQ_THRESHOLD_ETH_PORT_PFC`	Set PFC threshold value for ethernet port

Table 22: Threshold types

bcmCosqThresholdBytes	***consumed bytes. Egress Drop threshold, Unicast, multicast, dp (multicast) at port and device level. Egress Flow control threshold, unicast, multicast, port, channel and device level.***
bcmCosqThresholdPacketDescriptorsMin	Fair Adaptive Thresholds support for bcmCosqThresholdPacketDescriptors.
bcmCosqThresholdPacketDescriptorsMax	Fair Adaptive Thresholds support for bcmCosqThresholdPacketDescriptors.
bcmCosqThresholdPacketDescriptorsAlpha	Fair Adaptive Thresholds support for bcmCosqThresholdPacketDescriptors.
bcmCosqThresholdDataBuffersMin	Fair Adaptive Thresholds support for bcmCosqThresholdDataBuffers.
bcmCosqThresholdDataBuffersMax	Fair Adaptive Thresholds support for bcmCosqThresholdDataBuffers.
bcmCosqThresholdDataBuffersAlpha	Fair Adaptive Thresholds support for bcmCosqThresholdDataBuffers.

New GPORT types for `bcm_cosq_gport_handle_get()` API has been added.

Table 23: gport types

bcmCosqGportTypeFabricRxUnicast	***Fabric unicast FIFO***
bcmCosqGportTypeFabricRxMulticast	Fabric multicast FIFO
bcmCosqGportTypeFabricRxTdm	Fabric TDM FIFO
bcmCosqGportTypeFabricRxLocalUnicast	Local unicast FIFO
bcmCosqGportTypeFabricRxLocalMulticast	MESH-MC FIFO
bcmCosqGportTypeFabricRxLocalTdm	Local TDM bypass FIFO

New flags for new BCM CoSQ gport have been added.

Table 24: BCM CoSQ gport flags

Flags
BCM_COSQ_GPORT_SW_ONLY

The type of parameter core for `bcm_cosq_gport_handle_core_get()` API has been changed from `int` to `bcm_core_t.`

Two new flags for BCM COSQ Flow Control Endpoint configuration are added for `bcm_cosq_fc_path_add()` API.

Table 25: Flow Control Endpoint flags

BCM_COSQ_FC_PORT_OVER_PFC	*mapping PFC source to port target for FC reception*
BCM_COSQ_FC_INTF_COSQ_PFC	mapping PFC source to relevant prioroty in all ports on same interface for FC reception
BCM_COSQ_FC_ETH_PORT	FC triggered by RX Port FIFO

A new field `vfi_index` has been added into data structure `bcm_cosq_classifier_t` with `BCM_COSQ_CLASSIFIER_VFI` flag for `bcm_cosq_classifier_t_init()` API.

```
    typedef struct bcm_cosq_classifier_s {
        ...
        uint32 vfi_index;    /* VFI index */
    } bcm_cosq_classifier_t;
```

Table 26: BCM_COSQ_CLASSIFIER Flags

Name	*Purpose*
BCM_COSQ_CLASSIFIER_VFI	Classifier consists of VFI index

Below new APIs for COSQ have been added for this release.

bcm_cosq_congestion_mapping_info_t_init

Initialize a CoSQ congestion mapping information structure.

Syntax

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

Parameters

`config`	(IN/OUT) Congestion mapping configuration

Description

Initialize `bcm_cosq_congestion_mapping_info_t` data structure.

Returns

None.

bcm_cosq_congestion_mapping_set
bcm_cosq_congestion_mapping_get

Set/Get flow control bits as per valid cos values for the connected fabric module.

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_congestion_mapping_set(
int unit,
int fabric_modid,
bcm_cosq_congestion_mapping_info_t *mapping_info);
int bcm_cosq_congestion_mapping_get(
int unit,
int fabric_modid,
bcm_cosq_congestion_mapping_info_t *mapping_info);
```

Parameters

`unit`	(IN) BCM device number
`fabric_modid`	(IN) Fabric Module ID for which congestion data need to interpreted differently
`mapping_info`	(IN/OUT) Re-mapping Congestion mapping configuration

Description

These API's sets and gets the congestion state as per flow control bits specificid in `mapping_info` for the fabric module connected to the ingress module

```
    typedef struct bcm_cosq_congestion_mapping_info_s {
        bcm_cos_t flow_control_bits[BCM_COS_COUNT]; /* Valid flow control bits
in Fabric
                              0(Invalid), 1(Valid), -1(ignore). */
    } bcm_cosq_congestion_mapping_info_t;
```

Returns

bcm_cosq_ingress_queue_bundle_gport_add

Create an ingress queue bundle.

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_ingress_queue_bundle_gport_add(
    int unit,
    bcm_cosq_ingress_queue_bundle_gport_config_t *config,
    bcm_gport_t *gport);
```

Parameters

`unit`	(IN) Unit number.
`physical_port`	(IN) GPORT ID physical port
`config`	(IN) Config parameters for ingress queue bundle creation.
`gport`	(IN/OUT) GPORT ID for Queue group that is allocated.

Description

See description of `bcm_cosq_gport_add()` API.

Returns

bcm_cosq_src_vsqs_gport_add
bcm_cosq_src_vsqs_gport_get

Create the source based VSQ.

Syntax

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

int bcm_cosq_src_vsqs_gport_add(
    int unit,
    bcm_gport_t port,
    bcm_cosq_src_vsqs_gport_config_t *config,
    bcm_gport_t *src_port_vsq_gport,
    bcm_gport_t *pg_base_vsq_gport);
```

```
int bcm_cosq_src_vsqs_gport_get(
    int unit,
    bcm_gport_t port,
    bcm_cosq_src_vsqs_gport_config_t *config,
    bcm_gport_t *src_port_vsq_gport,
    bcm_gport_t *pg_base_vsq_gport);
```

Parameters

`unit`	[IN] BCM device number
`port`	[IN] Source port
`config`	[IN] (for `"_add")` Config parameters for src based VSQs attributes.
`src_port_vsq_gport`	[IN] (for `"_add")` source port VSQ gport
`pg_base_vsq_gport`	[IN] (for `"_add")` PG VSQ gport
`config`	[OUT] (for `"_get")` Config parameters for src based VSQs attributes.
`src_port_vsq_gport`	[OUT] (for `"_get")` source port VSQ gport
`pg_base_vsq_gport`	[OUT] (for `"_get")` PG VSQ gport

Returns

bcm_cosq_control_range_set
bcm_cosq_control_range_get

Set/Get a range configuration.

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_control_range_set(
    int unit,
    bcm_gport_t port,
    uint32 flags,
    bcm_cosq_control_range_type_t type,
    bcm_cosq_range_t *range);

int bcm_cosq_control_range_get(
    int unit,
    bcm_gport_t port,
    uint32 flags,
    bcm_cosq_control_range_type_t type,
    bcm_cosq_range_t *range);
```

Parameters

unit	(IN) BCM device number
port	(IN) A handle for gport to be configure.
flags	(IN) Flags
type	(IN) Range configuration type.
range	(IN) (for "_set") Range to be set.
range	(OUT) (for "_get") Range configured.

Description

Set/Get a range configuration.

Returns

cosq_resource_allocation_set cosq_resource_allocation_get

Set/Get the resource allocation for a target/entity

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_resource_allocation_set(
    int unit,
    uint32 flags,
    bcm_cosq_allocation_resource_t resource,
    bcm_cosq_allocation_entity_t *target,
    bcm_cosq_resource_amounts_t *amounts);

int bcm_cosq_resource_allocation_get(
    int unit,
    uint32 flags,
    bcm_cosq_allocation_resource_t resource,
    bcm_cosq_allocation_entity_t *target,
    bcm_cosq_resource_amounts_t *amounts);
```

Parameters

`unit`	(IN) BCM device number
`flags`	(IN) Flags
`resource`	(IN) Resource to allocate.
`target`	(IN) For which entity do we allocate/get the allocation.
`amounts`	(IN) (for `"_set")` The resource amounts to be allocated.
`amounts`	(OUT) (for `"_get")` The allocated amounts.

Description

Set/Get the resource allocation for a target/entity

Returns

PORT EXTENSION MANAGEMENT

A new Port Extension Data Type has been added in this release.

Table 27: BCM_EXTENDER_PORT flags

Name	*Purpose*
`BCM_EXTENDER_PORT_EGRESS_TRANSPARENT`	Use the flag to transmit packets transparently out of this extender port. It means ETAG of the packets out of this extender port will not be modified.
`BCM_EXTENDER_PORT_DROP`	Drop egress packets.
`BCM_EXTENDER_PORT_ID_ASSIGN_DISABLE`	Matched traffic is not assigned to the extender port.

FABRIC

Two fabric link control types have been added for `bcm_fabric_link_control_set/get()` APIs.

Table 28: bcm_fabric_link_control_t

Control Type	*Description*	*Arg Expected Values*
bcmFabricLinkTxTrafficDisable	Enable/disable Sending cells over specific link	0: Disable or 1: Enable
bcmFabricLinkRepeaterEnable	Enable/disable repeater link - enable means that the link is connected to repeater device	0: Disable or 1: Enable

One fabric link threshold type has been added for `bcm_fabric_link_thresholds_get/set()` APIs.

Table 29: bcm_fabric_link_threshold_type_t

enum	Description
bcmFabricLinkMidFull	Configure threshold for max fifo size, beyond this threshold, packets will be dropped.

New flag `BCM_FABRIC_PRIORITY_TDM` has been added into the available flags for `bcm_fabric_priority_set/get()` APIs.

Below new APIs for Fabric have been added in this release.

bcm_fabric_destination_link_min_set
bcm_fabric_destination_link_min_get

Configure minimum number of links per destination device.

Syntax

```
#include <bcm/fabric.h>
int
bcm_fabric_destination_link_min_set(
    int unit,
    uint32 flags,
    bcm_module_t module_id,
    int num_of_links)

int
bcm_fabric_destination_link_min_get(
    int unit,
    uint32 flags,
    bcm_module_t module_id,
    int *num_of_links)
```

Parameters

`unit`	(IN) Unit number
`flags`	(IN) Additional flags
`module_id`	(IN) Id of destination device
`int`	(IN/OUT) Nu of links to configure

Description

This functions are used to configure minimum number of links per destination device. Once the number of active links of a certain destination is lower than this value, all of its cells will be dropped.

Table 30: Configure all-reachable min number of links flag

Name	Purpose
`BCM_FABRIC_DESTINATION_LINK_MIN_ALL_REACHABLE`	configure all-reachable min number of links

Returns

bcm_fabric_rci_config_set
bcm_fabric_rci_config_get

configure RCI related information

Syntax

```
#include <bcm/fabric.h>
bcm_fabric_rci_config_set(
    int unit,
    bcm_fabric_rci_config_t rci_config)

bcm_fabric_rci_config_get(
    int unit,
    bcm_fabric_rci_config_t *rci_config)
```

Parameters

unit	(IN) Unit number
rci_config	(IN/OUT) Configuration struct

Description

This function is used to configure all related information regarding fabric RCI. all information will be configured / retrived to the configuration struct `bcm_fabric_rci_config_t. BCM_FABRIC_NUM_OF_RCI_LEVELS` is defined to 7 while `BCM_FABRIC_NUM_OF_RCI_SEVERITIES` is defined to 3.

```
    /* Fabric RCI configuration */
    typedef struct bcm_fabric_rci_config_s {
             /* thresholds for severity counter of RCI - per core */
         uint32 rci_core_level_thresholds[BCM_FABRIC_NUM_OF_RCI_LEVELS];
             /* thresholds for severity counter of RCI - per device */
         uint32 rci_device_level_thresholds[BCM_FABRIC_NUM_OF_RCI_LEVELS];
             /* RCI severity level factor */
         uint32 rci_severity_factors[BCM_FABRIC_NUM_OF_RCI_SEVERITIES];
             /* RCI score of fabric fifo */
         uint32 rci_high_score_fabric_rx_queue;
             /* RCI score of fabric local fifo */
         uint32 rci_high_score_fabric_rx_local_queue;
             /* number of congested links to be used as treshold */
         uint32 rci_threshold_num_of_congested_links;
             /* high score to be used when number of congested links reaches
threshold */
         uint32 rci_high_score_congested_links;
    } bcm_fabric_rci_config_t;
```

Returns

bcm_fabric_rci_config_t_init

Initialize a bcm_fabric_rci_config_t to a 'safe' default value.

Syntax

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

void
bcm_fabric_rci_config_t_init(bcm_fabric_rci_config_t *rci_config)
```

Parameters

`rci_config`	(OUT) Pointer to information struct to initialize

Description

Initialize a `bcm_fabric_rci_config_t` to a 'safe' default value.

Returns

void

FIELD PROCESSOR

Below BCM field qualifiers as well as the actions have been added in this release.

Table 31: New Field Qualifiers

Qualifier	*Purpose*
bcmFieldQualifyIngressInterfaceClassVPort	Class Id assigned for packet based on Ingress Virtual Port
bcmFieldQualifySrcVlanGport	Source Vlan gport.
bcmFieldQualifyDstVlanGport	Destination Vlan gport.
bcmFieldQualifySrcVlanGports	Source Vlan gports.
bcmFieldQualifyDstVlanGports	Destination Vlan gports.
bcmFieldQualifySrcVxlanGports	Source Vxlan gports.
bcmFieldQualifyDstVxlanGports	Destination Vxlan gports.
bcmFieldQualifySrcWlanGports	Source Wlan gports.
bcmFieldQualifyDstWlanGports	Destination Wlan gports.
bcmFieldQualifySrcMplsGports	Source Mpls gports.
bcmFieldQualifyDstMplsGports	Destination Mpls gports.
bcmFieldQualifySrcGports	Source module/port gport or MPLS/MiM/WLAN/Niv gports.
bcmFieldQualifyDstGports	Destination Niv gports.
bcmFieldQualifySrcMimGports	Source Mim gports.
bcmFieldQualifyDstMimGports	Destination Mim gports.
bcmFieldQualifySrcNivGports	Source Niv gports.

Table 31: New Field Qualifiers

Qualifier	Purpose
bcmFieldQualifyDstNivGports	Destination Niv gports.
bcmFieldQualifySrcModPortGports	Source module/port pair, regardless of trunk membership.
bcmFieldQualifyStackingRoute	Stacking Route
bcmFieldQualifyRxTrapStrength	Rx Trap Strength
bcmFieldQualifyVxlanHeaderBits8_31	Reserved_1 field in Vxlan Header.
bcmFieldQualifyVxlanHeaderBits56_63	Reserved_2 field in Vxlan Header.

Table 32: New Field Actions

Action	Description	param0	param1
bcmFieldActionPphPresentSet	If set, a Packet Processing header is present.	n/a	n/a
bcmFieldActionStackingRouteNew	Set the stacking route	n/a	n/a
bcmFieldActionVxlanHeaderBits8_31_Set	Set Vxlan Header Reserved_1 Field	Vxlan Reserved_1 field	n/a
bcmFieldActionVxlanHeaderBits56_63_Set	Set Vxlan Header Reserved_2 Field	Vxlan Reserved_2 field	n/a
bcmFieldActionFabricHeaderSet	If set, a Packet Processing header is present.	n/a	n/a

New enumerated defines for fabric headers are listed as following.

Table 33: bcm_field_fabric_header_t

heading Defines	heading Description
bcmFieldFabricHeaderEthernet	Ethernet fabric header
bcmFieldFabricHeaderTrafficManagementUcast	Unicast Traffic Management fabric header
bcmFieldFabricHeaderTrafficManagementMcast	Multiicast Traffic Management fabric header
bcmFieldFabricHeaderTrafficManagementVPort	VPort Traffic Management fabric header
bcmFieldFabricHeaderStacking	Stacking fabric header

New Field APIs in this release have been added as following.

bcm_field_qualify_XXX

Add a qualification to a field entry

Syntax

```
#include <bcm/field.h>
int bcm_field_qualify_RxTrapCode32(int unit, bcm_field_entry_t entry,
   uint32 data, uint32 mask);
int bcm_field_qualify_RxTrapStrength(int unit, bcm_field_entry_t entry,
     uint32 data, uint32 mask);
int bcm_field_qualify_LearnInVPort(int unit, bcm_field_entry_t entry,
   uint32 data, uint32 mask);
int bcm_field_qualify_StackingRoute(int unit, bcm_field_entry_t entry,
    uint32 data, uint32 mask);
int bcm_field_qualify_SrcVlanGport(int unit, bcm_field_entry_t entry,
   bcm_gport_t vlan_port_id);
int bcm_field_qualify_DstVlanGport(int unit, bcm_field_entry_t entry,
   bcm_gport_t vlan_port_id);
int bcm_field_qualify_IngressInterfaceClassVPort(int unit,
       bcm_field_entry_t entry,
       uint32 data, uint32 mask);
int bcm_field_qualify_SrcVlanGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t vlan_port_id, bcm_gport_t vlan_port_mask);

int bcm_field_qualify_DstVlanGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t vlan_port_id, bcm_gport_t vlan_port_mask);
int bcm_field_qualify_SrcVxlanGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t vxlan_port_id, bcm_gport_t vxlan_port_mask);
int bcm_field_qualify_DstVxlanGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t vxlan_port_id, bcm_gport_t vxlan_port_mask);
int bcm_field_qualify_SrcWlanGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t wlan_port_id, bcm_gport_t wlan_port_mask);
int bcm_field_qualify_DstWlanGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t wlan_port_id, bcm_gport_t wlan_port_mask);
int bcm_field_qualify_SrcMplsGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t mpls_port_id, bcm_gport_t mpls_port_mask);
int bcm_field_qualify_DstMplsGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t mpls_port_id, bcm_gport_t mpls_port_mask);
int bcm_field_qualify_SrcGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t port_id, bcm_gport_t port_mask);
int bcm_field_qualify_DstGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t port_id, bcm_gport_t port_mask);
int bcm_field_qualify_SrcMimGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t mim_port_id, bcm_gport_t mim_port_mask);
int bcm_field_qualify_DstMimGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t mim_port_id, bcm_gport_t mim_port_mask);
int bcm_field_qualify_SrcNivGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t niv_port_id, bcm_gport_t niv_port_mask);
int bcm_field_qualify_DstNivGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t niv_port_id, bcm_gport_t niv_port_mask);
int bcm_field_qualify_SrcModPortGports(int unit, bcm_field_entry_t entry,
       bcm_gport_t port_id, bcm_gport_t port_mask);
```

Parameters

`unit`	BCM device number
`entry`	Field entry ID
`data`	Data to match against
`mask`	Mask to choose which bits of data to match against

Description

Adds a qualification to a filter entry. Each qualification added makes the entry more specific and match fewer possible packets.

Returns

bcm_field_qualify_XXX_get

Get a qualification match criteria from a field entry

Syntax

```
#include <bcm/field.h>
int bcm_field_qualify_RxTrapCode32_get(int unit, bcm_field_entry_t entry,
       uint32 *data, uint32 *mask);
int bcm_field_qualify_RxTrapStrength_get(int unit, bcm_field_entry_t entry,
 uint32 *data, uint32 *mask);
int bcm_field_qualify_LearnInVPort_get(int unit, bcm_field_entry_t entry,
       uint32 *data, uint32 *mask);
int bcm_field_qualify_StackingRoute_get(int unit, bcm_field_entry_t entry,
uint32 *data, uint32 *mask);
int bcm_field_qualify_SrcVlanGport_get(int unit, bcm_field_entry_t entry,
       bcm_gport_t *vlan_port_id);
int bcm_field_qualify_DstVlanGport_get(int unit, bcm_field_entry_t entry,
       bcm_gport_t *vlan_port_id);
int bcm_field_qualify_IngressInterfaceClassVPort_get(int unit,
       bcm_field_entry_t entry,
       uint32 *data, uint32 *mask);
int bcm_field_qualify_SrcVlanGports_get(int unit, bcm_field_entry_t entry,
       bcm_gport_t *vlan_port_id, bcm_gport_t *vlan_port_mask);
int bcm_field_qualify_DstVlanGports_get(int unit, bcm_field_entry_t entry,
       bcm_gport_t *vlan_port_id, bcm_gport_t *vlan_port_mask);
int bcm_field_qualify_SrcVxlanGports_get(int unit, bcm_field_entry_t entry,
       bcm_gport_t *vxlan_port_id, bcm_gport_t *vxlan_port_mask);
int bcm_field_qualify_DstVxlanGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *vxlan_port_id, bcm_gport_t *vxlan_port_mask);
int bcm_field_qualify_SrcWlanGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *wlan_port_id, bcm_gport_t *wlan_port_mask);
int bcm_field_qualify_DstWlanGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *wlan_port_id, bcm_gport_t *wlan_port_mask);
int bcm_field_qualify_SrcMplsGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *mpls_port_id, bcm_gport_t *mpls_port_mask);
int bcm_field_qualify_DstMplsGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *mpls_port_id, bcm_gport_t *mpls_port_mask);
int bcm_field_qualify_SrcGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *port_id, bcm_gport_t *port_mask);
int bcm_field_qualify_DstGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *port_id, bcm_gport_t *port_mask);
int bcm_field_qualify_SrcMimGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *mim_port_id, bcm_gport_t *mim_port_mask);
int bcm_field_qualify_DstMimGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *mim_port_id, bcm_gport_t *mim_port_mask);
int bcm_field_qualify_SrcNivGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *niv_port_id, bcm_gport_t *niv_port_mask);
int bcm_field_qualify_DstNivGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *niv_port_id, bcm_gport_t *niv_port_mask);
int bcm_field_qualify_SrcModPortGports_get(int unit, bcm_field_entry_t entry,
        bcm_gport_t *port_id, bcm_gport_t *port_mask);
```

Parameters

`unit`	BCM device number
`entry`	Field entry ID
`data`	Data to match against
`mask`	Mask to choose which bits of data to match against

Description

Get a match criteria for a specific qualifier from a field entry.

KERNEL NETWORK (KNET) CONFIGURATION

A new Rx Packet Filter Destination Type `BCM_KNET_DEST_T_CALLBACK` for KNET has been added.

Table 34: New BCM KNET Rx Packet Filter Destination Types

Name	***Purpose***
`BCM_KNET_DEST_T_CALLBACK`	Invoke filter callback function

KERNEL CALLBACK FUNCTIONS

The KNET module supports two types of callback functions at the kernel level: packet callbacks and filter callbacks. The callback functions are shared across all switch devices in the system, and only one callback fuinction of each type can be installed, for example it is not possible two install two different filter callback functions.

Packet Callbacks

There are two types of packet callbacks, Rx callbacks and Tx callbacks.

The Rx callback is invoked after the CRC and optionally the VLAN tag have been stripped off the Rx packet, but before any (optional) RCPU header is added. Both the packet data and the associated meta data (DCB for XGS devices) is passed to the callback function.

The Tx callback function is invoked immediately after the packet is handed off to the switch DMA engine, i.e. after an optional RCPU header has been stripped and after a VLAN tag has been added. As for Rx packets, the callback function may modify both Tx packet data and meta data.

Both the Rx and Tx callback functions may return a `NULL` pointer to indicate that a packet should be dropped.

Use the following APIs to register and unregister a packet callback function:

```
#include <bcm-knet.h>

typedef struct sk_buff *
(*knet_skb_cb_f)(struct sk_buff *skb, int dev_no, void *meta);

int
bkn_rx_skb_cb_register(knet_skb_cb_f rx_cb);

int
bkn_rx_skb_cb_unregister(knet_skb_cb_f rx_cb);

int
bkn_tx_skb_cb_register(knet_skb_cb_f tx_cb);

int
bkn_tx_skb_cb_unregister(knet_skb_cb_f tx_cb);
```

Filter Callbacks

The purpose of the filter callback function is to perform specialized packet matching which is either impossible or inefficient to do with the standard KNET packet filters.

The filter callback function is called whenever a KNET packet filter with destination type `BCM_KNET_DEST_T_CALLBACK` is matched. This means that the filter callback can be inserted (even multiple times) in the prioritized list of packets filters. The filter callback should return a match/no-match result and update the destination type in the `kcom_filter_t` structure which is passed to the callback function.

Use the following APIs to register and unregister a filter callback function:

```
#include <bcm-knet.h>

typedef int
(*knet_filter_cb_f)(uint8_t *pkt, int size, int dev_no, void *meta,
            int chan, kcom_filter_t *filter);
```

```
int
bkn_filter_cb_register(knet_filter_cb_f filter_cb);

int
bkn_filter_cb_unregister(knet_filter_cb_f filter_cb);
```

CALLBACK EXAMPLES

The SDK source code contains a simple example implementation of both KNET packet callbacks and filter callbacks in:

```
$SDK/systems/linux/kernel/modules/knet-cb/knet-cb.c
```

LAYER 3 MANAGEMENT

A L3 flag `BCM_BCM_L3_ENCAP_SPACE_OPTIMIZED` for Optimize encapsulation database resources is defined and equivalent to `BCM_L3_RPE` for Pick up new priority (COS). Another `BCM_L3_CASCADED` flag is also added in this release.

Two new BCM L3 ECMP Dynamic Load Balancing Mode Flags have been added.

Table 35: BCM L3 ECMP Dynamic Load Balancing Mode Flags

Name	*Purpose*
`BCM_L3_ECMP_DYNAMIC_MODE_RANDOM`	ECMP randomized load balancing mode.
`BCM_L3_ECMP_DYNAMIC_MODE_ROUND_ROBIN`	ECMP round robin load balancing mode.

MAC-IN-MAC MANAGEMENT

A new MAC-in-MAC port flag `BCM_MIM_PORT_DEFAULT` has been added for `bcm_mim_port_add()` API.

Table 36: MAC-in-MAC Port Flags

Name	*Purpose*
`BCM_MIM_PORT_DEFAULT`	Used to indicate MiM default port which is used to forward MiM lookup missed packets.

MIRRORING

The new BCM Mirroring flag `BCM_MIRROR_PORT_SFLOW` has been added for `bcm_mirror_port_set()` API. In addition, the new BCM Mirror Destination flag `BCM_MIRROR_DEST_TUNNEL_SFLOW` has been added while two fields `udp_src_port` and `udp_dst_port` have been added in `bcm_mirror_destination_t` data structure.

Table 37: BCM Mirroring Flags

Name	Description
BCM_MIRROR_PORT_SFLOW	sFlow mirroring

Table 38: BCM Mirror Destination Flags

Name	Description
BCM_MIRROR_DEST_TUNNEL_SFLOW	Mirrored packet should be with sFlow encapsulation

```
/*
 * Mirror destination Structure
 *
 * Contains information required for manipulating mirror destinations.
 */
typedef struct bcm_mirror_destination_s {
    ...
    uint16 udp_src_port;                /* UDP source port */
    uint16 udp_dst_port;                /* UDP destination port */
    ...
} bcm_mirror_destination_t;
```

MPLS MANAGEMENT

A new extension of MPLS port flag BCM_MPLS_PORT2_LEARN_ENCAP has been added.

Table 39: MPLS Port Flags(2)

Name	Purpose
BCM_MPLS_PORT2_LEARN_ENCAP	Learning information is a pointer to encapsulation database entry
BCM_MPLS_PORT2_CASCADED	Set in case the mpls port is over overlay

OPERATIONS, ADMINISTRATION, AND MAINTENANCE

A new endpoint2 flag has been added in this release.

A new Loss Measurement flag has been added in this release.

Table 40: New OAM Loss Measurement Flag Definitions

Flag	Description
BCM_OAM_LOSS_REPORT_MODE	Enable report mode.

A new Delay Measurement flag has been added in this release.

Table 41: OAM Delay Measurement Flag Definitions

Flag	Description
BCM_OAM_DELAY_REPORT_MODE	Enable report mode.

New OAM Event types have been added in this release.

Table 42: OAM Event Types

Event type	Description
bcmOAMEventEndpointDmStatistics	Delay measurement statistics (performance event only)
bcmOAMEventEndpointLmStatistics	Loss measurement statistics (performance event only)

NEW OAM AIS (ALARM INDICATION SIGNAL) OBJECTS ARE DEFINED

```
/* OAM AIS (alarm indication signal) object. */
typedef struct bcm_oam_ais_s {
    uint32 flags;
    int ais_id;                      /* OAM AIS ID. */
    bcm_oam_endpoint_t id;           /* Endpoint ID of Local MEP. */
    int period;                      /* Frequency of AIS tx period in
                       milliseconds. */
    int level;                       /* MD level on outgoing frames. */
    bcm_mac_t peer_da_mac_address;   /* MAC DA in AIS injection. */
} bcm_oam_ais_t;
```

Table 43: OAM AIS Flag Definitions

Flag	Description
BCM_OAM_AIS_UPDATE	Update AIS settings for given endpoint
BCM_OAM_AIS_MULTICAST	Transmit AIS frames with multicasat DA address
BCM_OAM_AIS_WITH_AIS_ID	use given ais_id
BCM_OAM_AIS_TRANSMIT_START	Enable AIS transmission for given local endpoint
BCM_OAM_AIS_TRANSMIT_STOP	Disable AIS transmission for given local endpoint

The below callback function is defined for supplying function pointer for the registration and unregistration APIs to handle performance event with the data type bcm_oam_performance_event_data_t.

```
typedef int (*bcm_oam_performance_event_cb)(
    int unit,
    bcm_oam_event_type_t event_type,
    bcm_oam_group_t group,
    bcm_oam_endpoint_t endpoint,
    bcm_oam_performance_event_data_t *event_data,
    void *user_data);

typedef struct bcm_oam_performance_event_data_s {
   uint32 delta_FCB;   /* TXFcb - RXFcb. Used for performance events of type
```

```
                 bcmOAMEventEndpointLMStatistics. */
      uint32 delta_FCf;   /* TXFcf - RXFcb. Used for performance events of type
                 bcmOAMEventEndpointLMStatistics. */
       uint64 last_delay;  /* Used for performance events of type
                 bcmOAMEventEndpointDMStatistics. */
  } bcm_oam_performance_event_data_t;

       /* Initialize an performance event data structure */
       extern void bcm_oam_performance_event_data_t_init(
               bcm_oam_performance_event_data_t *performance_event_data_ptr);
```

Some other miscellaneous BFD defines are added.

Table 44: New miscellaneous OAM defines

Define Name	***Description***
BCM_OAM_ENDPOINT_DEFAULT_EGRESS1	OAM default endpoint, egress
BCM_OAM_ENDPOINT_DEFAULT_EGRESS2	OAM default endpoint, egress
BCM_OAM_ENDPOINT_DEFAULT_EGRESS3	OAM default endpoint, egress

New APIs for OAM are listed as following.

bcm_oam_performance_event_register

Register a handler for OAM performance events.

Syntax

```
#include <bcm/oam.h>
int bcm_oam_performance_event_register(
    int unit,
    bcm_oam_event_types_t event_types,
    bcm_oam_performance_event_cb cb,
    bcm_oam_performance_event_data_t *event_data,
    void *user_data);
```

Parameters

unit	BCM device number
event_types	(IN) A collection of event types for which the specified callback will be called (May only be one of the events defined to be performance events).
cb	(IN) Pointer to the callback function
event_data	(IN) Pointer to a Performance event data struct. When a performance event interrupt will be triggered this struct will be fill with the statistcs, depending on the event type and the call back will be called.
user_data	(OUT) Pointer to arbitrary user data to be passed back to the callback

Description

Registers the specified callback to handle the performance event types specified in the collection.

Returns

bcm_oam_performance_event_unregister

Unregister a handler for OAM performance events.

Syntax

```
#include <bcm/oam.h>
int bcm_oam_performance_event_unregister(
    int unit,
    bcm_oam_event_types_t event_types,
    bcm_oam_performance_event_cb cb);
```

Parameters

unit	BCM device number
event_types	(IN) A collection of event types for which the specified callback will no longer be called.
cb	(IN) Pointer to the callback function

Description

Unregisters the specified callback from handling the event types specified in the collection. The callback will continue to be called for any other event types for which it remains registered.

Returns

bcm_oam_ais_t_init

Initialize a bcm_oam_ais_t structure.

Syntax

```
#include <bcm/oam.h>
void bcm_oam_ais_t_init(bcm_oam_ais_t *ais_ptr);
```

Parameters

`ais_ptr`	(OUT) Pointer to OAM ais structure to initialize

Description

Initializes an OAM ais structure to default values. This function must be used to initialize any OAM ais structure before passing it to an API function.

bcm_oam_ais_add

Add an OAM ais object

Syntax

```
#include <bcm/oam.h>
int
bcm_oam_ais_add(
    int unit,
    bcm_oam_ais_t *ais_ptr);
```

Parameters

`unit`	(IN) Unit number.
`ais_ptr`	(IN) The OAM ais object to add

Description

Add an OAM ais object.

Returns

bcm_oam_ais_get

Get an OAM ais object

Syntax

```
#include <bcm/oam.h>
int
```

```
bcm_oam_ais_get(
    int unit,
    bcm_oam_ais_t *ais_ptr);
```

Parameters

unit	(IN) Unit number.
ais_ptr	(OUT) The OAM ais object to get

Description

Get an OAM ais object.

Returns

bcm_oam_ais_delete

Delete an OAM ais object

Syntax

```
#include <bcm/oam.h>
int
bcm_oam_ais_delete(
    int unit,
    bcm_oam_ais_t *ais_ptr);
```

Parameters

unit	(IN) Unit number.
ais_ptr	(IN) The OAM ais object to delete

Description

Delete an OAM ais object.

Returns

POLICER CONFIGURATION

A new field `pool_id` of data type `bcm_policer_config_t` has been added with the new policer flag `BCM_POLICER_WITH_POOL_ID` has been added for `bcm_policer_create()` API.

Table 45: Policer Flags

Name	*Purpose*
`BCM_POLICER_WITH_POOL_ID`	Policer needs to be created with the given PoolId

```
    typedef struct bcm_policer_config_s {
       ...
       bcm_policer_pool_t pool_id; /* Meter Pool Id of the current policer */
    } bcm_policer_config_t;
```

New API for policer are added as following.

bcm_policer_group_create_with_map

Allocate a block of policer entries with flexible mapping.
bcm_policer_set must be called to setup the individual policers.

Syntax

```
#include <bcm/policer.h>
int
bcm_policer_group_create_with_map(
    int unit,
    bcm_policer_group_mode_t mode,
    bcm_policer_offset_map *offset_map,
    bcm_policer_t *policer_id,
    int *npolicers);
```

Parameters

`unit`	(IN) Unit number.
`mode`	(IN) Policer Group Mode.
`offset_map`	(IN) Structure with mapping of offset to policer.
`policer_id`	(OUT) Base Policer ID.
`npolicers`	(IN/OUT) Number of Policers created including base Policer ID. In case of "bcmPolicerGroupModeCascade" and "bcmPolicerGroupModeCascadeWithCoupling" modes, this parameter indicates the number of policers to be created

Description

Allocates the requested number of policers and returns the base policer ID. This API does not set up the policers in the hardware. `bcm_policer_set()` must be called to setup the individual policers.

```
    typedef struct bcm_policer_map_s {
        int count;          /* Number of offset mappings */
        uint8 *offset_map;  /* policer offset for each index */
    } bcm_policer_map_t;
```

Returns

```
BCM_E_XXX
```

New APIs with the data types for advanced policer group creation have been added in this release as following.

ADVANCED POLICER GROUP CREATION

Existing policer group create API's allow creation of a group of policers based on pre-defined set of policer group modes. These group modes today come with inherent limitatios in terms of number of policers that need to be created in the group as well as in terms of which policer needs to be used for what kind of packet. There is no flexiblity in choosing the number of policers and how different packets map to different policers. Also, these API's lack the support for policer group creation based on UDF attributes. In order to overcome these limitation below API's are defined.

```
    /* Policer group mode types */
    typedef enum bcm_policer_group_mode_type_e {

       /* A group of n policers (max 256) which do not share bandwidth amongst
each other */
        bcmPolicerGroupModeTypeNormal = 0,

        /* User specified set of n policers (max 8), wherein excess bandwidth
can overflow
          from high priority policers (i.e policer with higher index)bucket to
low priority
           policers bucket(policer with next higher index in the group) */
        bcmPolicerGroupModeTypeCascade = 1,

       /* User specified set of policers (max 4),  wherein excess bandwidth can
overflow
          from high priority (i.e policer with higher index)policers bucket to
low priority
```

```
           bucket and from low priority green bucket to high priority yellow 
bucket */
        bcmPolicerGroupModeTypeCascadeWithCoupling = 2,

       /* Always last, Not to be used */
        bcmPolicerGroupModeTypeCount = 3
    } bcm_policer_group_mode_type_t;


    /* Vlan related policer attributes values */
    typedef enum bcm_policer_group_mode_attr_vlan_e {
        bcmPolicerGroupModeAttrVlanUnTagged =   1, /* UnTagged Vlan */
        bcmPolicerGroupModeAttrVlanInnerTag =   2, /* Inner Tagged Vlan */
        bcmPolicerGroupModeAttrVlanOuterTag =   3, /* Outer Tagged Vlan */
       bcmPolicerGroupModeAttrVlanStackedTag = 4, /* Both Inner & Outer Tagged 
Vlan */
      bcmPolicerGroupModeAttrVlanAll =      5  /* UnTagged,Inner & Outer Tagged 
                               Vlan */
    } bcm_policer_group_mode_attr_vlan_t;

    /* Packet type related policer attributes values */
    typedef enum bcm_policer_group_mode_attr_pkt_type_e {
        /* All Packet Types */
        bcmPolicerGroupModeAttrPktTypeAll =                 1,

        /* Unknown Packet */
        bcmPolicerGroupModeAttrPktTypeUnknown =             2,

        /* Control Packet */
        bcmPolicerGroupModeAttrPktTypeControl =             3,

        /* OAM Packet */
        bcmPolicerGroupModeAttrPktTypeOAM =                 4,

        /* BFD Packet */
        bcmPolicerGroupModeAttrPktTypeBFD =                 5,

        /* BPDU Packet */
        bcmPolicerGroupModeAttrPktTypeBPDU =                6,

        /* ICNM Packet */
        bcmPolicerGroupModeAttrPktTypeICNM =                7,

        /* 1588 Packet */
        bcmPolicerGroupModeAttrPktType1588 =                8,

        /* Known L2 Unicast Packet */
        bcmPolicerGroupModeAttrPktTypeKnownL2UC =           9,

        /* Unknown L2 Unicast Packet */
        bcmPolicerGroupModeAttrPktTypeUnknownL2UC =         10,

        /* L2 Broadcast Packet */
        bcmPolicerGroupModeAttrPktTypeL2BC =               11,

        /* Known L2 Multicast Packet */
        bcmPolicerGroupModeAttrPktTypeKnownL2MC =          12,

        /* Unknown L2 Multicast Packet */
        bcmPolicerGroupModeAttrPktTypeUnknownL2MC =        13,

        /* Known L3 Unicast Packet */
        bcmPolicerGroupModeAttrPktTypeKnownL3UC =          14,

        /* Unknown L3 Unicast Packet */
        bcmPolicerGroupModeAttrPktTypeUnknownL3UC =        15,

        /* Known IPMC Packet */
        bcmPolicerGroupModeAttrPktTypeKnownIPMC =          16,

        /* Unknown IPMC Packet */
        bcmPolicerGroupModeAttrPktTypeUnknownIPMC =        17,

        /* Known MPLS L2 Packet */
        bcmPolicerGroupModeAttrPktTypeKnownMplsL2 =        18,

        /* Known MPLS L3 Packet */
        bcmPolicerGroupModeAttrPktTypeKnownMplsL3 =        19,

        /* Known MPLS Packet */
        bcmPolicerGroupModeAttrPktTypeKnownMpls =          20,

        /* Unknown MPLS Packet */
        bcmPolicerGroupModeAttrPktTypeUnknownMpls =        21,

        /* Known MPLS Multicast Packet */
        bcmPolicerGroupModeAttrPktTypeKnownMplsMulticast = 22,

        /* Known MiM Packet */
        bcmPolicerGroupModeAttrPktTypeKnownMim =           23,

        /* Unknown MiM Packet */
        bcmPolicerGroupModeAttrPktTypeUnknownMim =         24,

        /* Known TRILL Packet */
        bcmPolicerGroupModeAttrPktTypeKnownTrill =         25,

        /* Unknown TRILL Packet */
        bcmPolicerGroupModeAttrPktTypeUnknownTrill =       26,

        /* Known  NIV Packet */
        bcmPolicerGroupModeAttrPktTypeKnownNiv =           27,

        /* Unknown  NIV Packet */
        bcmPolicerGroupModeAttrPktTypeUnknownNiv =         28
    } bcm_policer_group_mode_attr_pkt_type_t;
```

```
/* Policer Group Mode Attribute Selectors */
typedef enum bcm_policer_group_mode_attr_e {
   /*  UDF selector: Possible values:0-255. */
    bcmPolicerGroupModeAttrUdf =                        1,

   /*  Field Ingress Color Selector: Possible
       Values:bcmColorGreen|Yellow|REd or
       POLICER_GROUP_MODE_ATTR_ALL_VALUES for all */
    bcmPolicerGroupModeAttrFieldIngressColor =          2,

    /* Internal Priority Selector: Possible Values:
       0 to 15 or POLICER_GROUP_MODE_ATTR_ALL_VALUES
       for all */
    bcmPolicerGroupModeAttrIntPri =                     3,

    /* Vlan Type Selector: Possible Values:
       bcmPolicerGroupModeVlanAttr */
    bcmPolicerGroupModeAttrVlan =                       4,

    /* Outer Vlan Priority Selector:Possible Values:
       0 to 7 or POLICER_GROUP_MODE_ATTR_ALL_VALUES
       for all */
    bcmPolicerGroupModeAttrOuterPri =                   5,

    /* Inner Vlan Priority Selector:Possible Values:
       0 to 7 or POLICER_GROUP_MODE_ATTR_ALL_VALUES
       for all */
    bcmPolicerGroupModeAttrInnerPri =                   6,

    /* Logical Port Selector: Possible Values:
       <MaxLogicalPort> or
       POLICER_GROUP_MODE_ATTR_ALL_VALUES for all */
    bcmPolicerGroupModeAttrPort =                       7,

    /* Type Of Service Selector
       (DSCP : Differentiated services Code Point):
       Possible Values:<6b:TOS Val> or
       POLICER_GROUP_MODE_ATTR_ALL_VALUES for all */
    bcmPolicerGroupModeAttrTosDscp =                    8,

    /* Type Of Service Selector
       (ECN: Explicit Congestion Notification):
       Possible Values:<2b:TOS Val> or
       POLICER_GROUP_MODE_ATTR_ALL_VALUES for all */
    bcmPolicerGroupModeAttrTosEcn =                     9,

    /* Packet Type Selector: Possible Values:
       <bcmPolicerGroupModeAttrPktType*> */
    bcmPolicerGroupModeAttrPktType =                    10,

    /* Ingress Network Group Selector:
       Possible Values:<Value> or
```

```
           POLICER_GROUP_MODE_ATTR_ALL_VALUES for all */
        bcmPolicerGroupModeAttrIngNetworkGroup =             11,

        bcmPolicerGroupModeAttrDrop =                        12,
        /* Drop Selector:
           Possible Values:<0 or 1> */

        /* Ip Packet Selector:
           Possible Values:<0 or 1> */
        bcmPolicerGroupModeAttrPacketTypeIp =                13

       /* Last value. Not to be used */
          bcmPolicerGroupModeAttrCount =                     14
    } bcm_policer_group_mode_attr_t;

    /* Unique constants */

    /* A unique constant for using all attr values */
    #define BCM_POLICER_GROUP_MODE_ATTR_ALL_VALUES 0xFFFFFFFF

    /* packet attribute selector flags values */

    /* Offset and width fields are valid for this attribute */
    #define BCM_POLICER_ATTR_WIDTH_OFFSET   0x1



    /* Policer Group Attribute Selector */
    typedef struct bcm_policer_group_mode_attr_selector_s {
       uint32 flags;                       /* packet attribute selector flags */
       uint32 policer_offset;               /* Policer Offset */
       bcm_policer_group_mode_attr_t attr; /* Attribute Selector */
       uint32 attr_value;                   /* Attribute Values */
      int udf_id;                       /* UDF ID created using bcm_udf_create.
Applicable only if Group mode attribute is bcmPolicerGroupModeAttrUdf. */
       uint32 offset;                      /* Offset in bits from the beginning
of the packet attribute */
       uint32 width;                      /* Number of bits in packet attribute
from offset to be used for policer offset mapping */
    } bcm_policer_group_mode_attr_selector_t;
```

bcm_policer_group_mode_id_create

Create Customized Policer Group mode for given Policer Attributes

Syntax

```
#include <bcm/policer.h>
int
bcm_policer_group_mode_id_create(
    int unit,
    uint32 flags,
    bcm_policer_group_mode_type_t type,
    uint32 total_policers,
    uint32 num_selectors,
    bcm_policer_group_mode_attr_selector_t *attr_selectors,
    uint32 *mode_id)
```

Parameters

`unit`	(IN) Unit number.
`flags`	(IN) flags
`type`	(IN) various modes of policer group creation - normal/cascade/cascade with coupling.
`total_policers`	(IN) Total policers for Policer Group Mode
`num_selectors`	(IN) Number of Selectors for Policer Group Mode
`attr_selectors`	(IN) Attribute Selectors for Policer Group Mode
`mode_id`	(OUT) Created Mode Identifier for Policer Group Mode

Description

This API create Customized Policer Group mode for given Policer Attributes.

Returns

```
BCM_E_XXX
```

bcm_policer_group_mode_id_get

Retrieves Customized Policer Group mode Attributes for given mode_id

Syntax

```
#include <bcm/policer.h>
int
bcm_policer_group_mode_id_get(
    int unit,
    int mode_id,
    uint32 num_selectors,
    uint32 *flags,
    bcm_policer_group_mode_type_t *type,
    uint32 *total_policers,
    bcm_policer_group_mode_attr_selector_t *attr_selectors,
    uint32 *actual_num_selectors)
```

Parameters

`unit`	(IN) Unit number.
`mode_id`	(IN) Created Mode Identifier for Policer GroupMode
`num_selectors`	(IN) Number of Selectors for Stat Flex Group Mode
`flags`	(OUT) flags
`total_policers`	(OUT) Total Policers in Policer Group Mode
`attr_selectors`	(OUT) Attribute Selectors for Policer Group Mode
`actual_num_selectors`	(OUT) Actual Number of Selectors for Policer GroupMode

Description

This API retrieves Customized Policer Group mode Attributes for given mode id

Returns

```
BCM_E_XXX
```

bcm_policer_group_mode_id_destroy

Destroys Customized Group mode

Syntax

```
#include <bcm/policer.h>
int
bcm_policer_group_mode_id_destroy(
    int unit,
    uint32 mode_id)
```

Parameters

`unit`	(IN) Unit number.
`mode_id`	(IN) Created Mode Identifier for Policer Group Mode

Description

This API Destroys created Customized Group mode

Returns

```
BCM_E_XXX
```

bcm_policer_custom_group_create

Create a group of policers belonging to group mode specified

Syntax

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

```
int
bcm_policer_custom_group_create(
    int unit,
    uint32 flags,
    uint32 mode_id,
    bcm_policer_t macro_flow_policer_id,
    bcm_policer_t *policer_id,
    int *npolicers);
```

Parameters

`unit`	(IN) Unit number.
`flags`	(IN) flags
`mode_id`	(IN) Created Mode Identifier for Policer Group Mode
`macro_flow_policer_id`	(IN) is an optional parameter and needs to be passed only while creating the micro flow policers of a hierarchical group (2 stage policers).
`policer_id`	(OUT) base policer ID
`npolicers`	(OUT) Number of policer entries created

Description

This API creates a group of policers belonging to customized group mode. `bcm_policer_set` must be called to setup the individual policers. The parameter `macro_flow_policer_id` needs to be set to "0" in case of 1 level policers. In case of 2 level policers, this paramter needs to be set with the value of second level policer while creating 1st level policers.

Returns

`BCM_E_XXX`

PORT CONFIGURATION

Some new speeds are added in the extended abilities. These abilities are also determined during the initialization of the port module. A common structure is used for retrieving port extended abilities. These are defined in `EXTENDED_PORT_ABILITY`.

Table 46: EXTENDED_PORT_ABILITY

BCM_PORT_ABILITY_27GB	***27 Gbps speed***
`BCM_PORT_ABILITY_50GB`	50 Gbps speed
`BCM_PORT_ABILITY_53GB`	53 Gbps speed

The port interface type `BCM_PORT_IF_CPU` has been added while port interface value `BCM_PORT_IF_XLAUI2` has been added in the values of Port Interface Table.

Table 47: Port interfaces

Flag	***Meaning***
`BCM_PORT_IF_CPU`	CPU interface (phy port must be 0).

A new port class has been added for set or get APIs of port classification ID to aggregate a group of ports for further processing such as VLAN translation and field processing.

New port control flags have been added for `bcm_port_control_get/set()` APIs.

Table 48: bcm_port_control_t

bcmPortControlTrunkLoadBalancingRandomizer	Assign Port-specific Randomizer-value to bias the random-member
bcmPortControlECMPLevel1LoadBalancingRandomizer	Assign Port-specific Randomizer-value to bias the random-member selection of ECMP(at Level 1 if Hierarchical ECMP Mode is configured). Use integer-value between 0 and 15.
bcmPortControlECMPLevel2LoadBalancingRandomizer	Assign Port-specific Randomizer-value to bias the random-member selection of ECMP at Level 2 in Hierarchical ECMP Mode. Not used in Single Level Mode. Use integer-value between 0 and 15.
bcmPortControlMplsExplicitNullEnable	Enable or disable Explicit `NULL` termination database. Use integer-value between 0 and 1.
bcmPortControlExtenderEnable	Enable/disable port extender capability on the port Use integer-value between 0 and 1.

New Port PHY control flags have been added for `bcm_port_phy_control_set/get()` APIs.

Table 49: bcm_port_phy_control_t

BCM_PORT_PHY_CONTROL_CLOCK_MODE_AUTO	Automatically disable the clock when link goes down.
`BCM_PORT_PHY_CONTROL_CLOCK_AUTO_SECONDARY`	Automatic switch primary recovery clock to secondary source when link failure.
`BCM_PORT_PHY_CONTROL_CLOCK_SOURCE`	Get the source of primary recovered clock at present. The returned values were defined in below `bcm_port_phy_clock_source_t` (page 52)

Table 50: bcm_port_phy_clock_source_t

Value	Description
bcmPortPhyClockSourcePrimary	Recovery clock is being derived from primary port's recovered clock
bcmPortPhyClockSourceSecondary	Recovery clock is being derived from the secondary port's recovered clock

Some Port congestion control flags have been added for `bcm_port_congestion_config_set/get()` APIs.

Table 51: Port Congestion flags

Name	Description
`BCM_PORT_CONGESTION_CONFIG_DESTMODPORT_FLOW_CONTROL`	To set, DPVOQ specific configuration
`BCM_PORT_CONGESTION_CONFIG_E2ECC_COE_STRICT_MODE`	To enable E2ECC FC on COE port in strict mode
`BCM_PORT_CONGESTION_CONFIG_E2ECC_COE_FLEX_MODE`	To enable E2ECC FC on COE port in flex mode

A new field extended_port_vid has been added for data structure bcm_port_match_info_t with its corresponding matching criterias.

```
    typedef struct bcm_port_match_info_s {
        ...
        uint16 extended_port_vid;           /* Extender port VID */
    } bcm_port_match_info_t;
```

Table 52: Generic Port Match Criteria

BCM_PORT_MATCH_PORT_UNTAGGED	Mod/port/trunk for untagged packets.
BCM_PORT_MATCH_PORT_EXTENDED_PORT_VID	Mod/port/trunk + Extender port VID.
BCM_PORT_MATCH_PORT_EXTENDED_PORT_VID_VLAN	Mod/port/trunk + Extender port VID + VLAN.

A new timesync configuration flag BCM_PORT_TIMESYNC_TIMESTAMP_CFUPDATE_ALL has been added for bcm_port_timesync_config_set/get() APIs.

New port APIs are listed as following.

bcm_port_add
bcm_port_get
bcm_port_remove

add/remove/get port

Syntax

```
#include <bcm/port.h>
int bcm_port_add(int unit, bcm_port_t port, uint32 flags,
bcm_port_interface_info_t *interface_info, bcm_port_mapping_info_t
*mapping_info);
int bcm_port_get(int unit, bcm_port_t port, uint32 *flags,
bcm_port_interface_info_t *interface_info, bcm_port_mapping_info_t
*mapping_info);
int bcm_port_remove(int unit, bcm_port_t port, uint32 flags) ;
```

Parameters

unit	(IN) BCM device number
port	(IN) Device or logical port number
flags	(IN) See BCM_PORT_ADD_flags (page 54) for details.
interface_info	(IN) (for "_set") The requested interface configuration
mapping_info	(IN) (for "_set") The requested port mapping configuration
interface_info	(OUT) (for "_get") The requested interface configuration
mapping_info	(OUT) (for "_get") The requested port mapping configuration

Description

Allows for dynamic NIF port configuration changes. Calling bcm_petra_port_add must be done on detached ports.

Table 53: bcm_port_interface_config_t

Field	Type	Description
phy_port	uint32	related physical port (first physical port for multi-lane interfaces)
interface	bcm_port_if_t	interface of port (See BCM_PORT_IF_types (page 55) table for supported interface types)

Table 54: bcm_port_mapping_info_t

Field	Type	Description
channel	uint32	related port channel
core	int	related port core id
core	uint32	tm_port corresponding tm port
core	uint32	base_q_pair corresponding base queue pair (relevant in case BCM_PORT_ADD_BASE_Q_PAIR_WITH_ID flag is set)
core	uint32	num_priorities ports number of priorities

Table 55: Port add Flags

Flag	Meaning
BCM_PORT_ADD_BASE_Q_PAIR_WITH_ID	The id of base queue pair is given in bcm_port_mapping_info_t strcuture.
BCM_PORT_ADD_CONFIG_CHANNELIZED	The added port if part of an channelized interface (relevent for non-Interlaken interfaces).
BCM_PORT_ADD_CGM_AUTO_ADJUST_DISABLE	disable CGM default adjustments, if set the user is responsible for CGM configuration.
BCM_PORT_ADD_HIGH_GIG	Indicate a high gig port.

Table 55: Port add Flags

Flag	Meaning
BCM_PORT_ADD_DONT_PROBE	Don't probe(initialize SerDes and MAC) the new physical interface.

Table 56: Port interfaces

Flag	Meaning
BCM_PORT_IF_SGMII	SGMII interface.
BCM_PORT_IF_CAUI	CAUI interface.
BCM_PORT_IF_RXAUI	RXAUI interface.
BCM_PORT_IF_XFI	XFI interface.
BCM_PORT_IF_XLAUI	XLAUI interface.
BCM_PORT_IF_ILKN	INTERLAKEN interface.
BCM_PORT_IF_XAUI	XAUI interface.
BCM_PORT_IF_CPU	CPU interface (phy port must be 0).

Returns

bcm_port_extender_mapping_info_t_init

Initialize a port extender mapping struct.

Syntax

```
#include <bcm/port.h>
void bcm_port_extender_mapping_info_t_init(bcm_port_extender_mapping_info_t
*mapping_info);
```

Parameters

mapping_info	(OUT) Pointer to the port extender mapping struct.

Description

Initializes a port extender mapping structure to default values. This function should be used to initialize any port extender mapping 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_port_extender_mapping_info_t structure, and bcm_port_extender_mapping_info_t will initialize the new members to correct default values.

Returns

None.

bcm_port_extender_mapping_info_set bcm_port_extender_mapping_info_get

Configure/retrieve port extender mapping from different inputs to Local pp port.

Syntax

```
#include <bcm/port.h>
int bcm_petra_port_extender_mapping_info_set(
    int unit,
    uint32 flags,
    bcm_port_extender_mapping_type_t type,
    bcm_port_extender_mapping_info_t *mapping_info);

int bcm_petra_port_extender_mapping_info_get(
    int unit,
    uint32 flags,
    bcm_port_extender_mapping_type_t type,
    bcm_port_extender_mapping_info_t *mapping_info);
```

Parameters

unit	(IN) Unit number.
flags	(IN) BCM_PORT_EXTENDER_MAPPING_XXXX
type	(IN) type of mapping info
mapping_info	(IN) (for "_set") mapping info (PON port+tunnel id/VLAN/user define value)
mapping_info	(INOUT) (for "_get") mapping info (PON port+tunnel id/VLAN/user define value)

Description

Configure/retrieve port extender mapping from different inputs to Local pp port.

Table 57: Port extender mapping type values

bcmPortExtenderMappingTypePonTunnel	*Map information includes PON port and Tunnel-ID*
bcmPortExtenderMappingTypePortVlan	Map information includes Phy Port and VLAN-ID
bcmPortExtenderMappingTypeUserDefineValue	Map information includes user define fields

Table 58: Port Extender Mapping flags

Name	*Description*
BCM_PORT_EXTENDER_MAPPING_INGRESS	Map information to Incoming local PP port
BCM_PORT_EXTENDER_MAPPING_EGRESS	Map information to Outgoing local PP port

Table 59: bcm_port_extender_mapping_info_t

Field	BCM Type	Description
pp_port	bcm_gport_t	Mapped Local PP port
tunnel_id	bcm_tunnel_id_t	Tunnel-ID
phy_port	bcm_gport_t	Physical Port connected to interface x channel
vlan	bcm_vlan_t	VLAN-ID
user_define_value	uint32	User define value from header

Returns

PRECISION TIME PROTOCOL

A new PTP TOD format, bcmPTPTODFormatBCMTS, has been added in this release

```
/* PTP TOD Formats */
typedef enum bcm_ptp_tod_format_e {
    bcmPTPTODFormatString,
    bcmPTPTODFormatUBlox,
    bcmPTPTODFormatChinaTcom,
    bcmPTPTODFormatBCM,
    bcmPTPTODFormatBCMTS
} bcm_ptp_tod_format_t;
```

In the PTP channel data strucutre bcm_ptp_channel_t, the field of source in bcm_ptp_timestamp_source_t has actually been the meaning of TSGPIO pin index or syncE port index.

QUALITY OF SERVICE

A new QoS configuration flag BCM_QOS_MAP_OAM_PCP has been added in this release.

RATE LIMITING

Two rate macros have been defined for the parameter kbits_sec of bcm_rate_bandwidth_get/set() APIs.

Table 60:

Macro	Meaning
BCM_RATE_DISABLE	Disables the metering
BCM_RATE_BLOCK	Blocks all packets @

PACKET TRANSMIT AND RECEIVE

A new field `forwarding_type` of data structure `bcm_rx_trap_config_t` along with the new trap code have been added for this release. New flags used in the trap/snoop configuration structure have also been defined.

Table 61: Rx Trap Codes.

Trap Code	***Description***
bcmRxTrapBfdEchoOverIpv4	IPv4 BFD Echo.

```
typedef struct bcm_rx_trap_config_s {
    ...
    bcm_forwarding_type_t forwarding_type; /* Forwarding type value */
} bcm_rx_trap_config_t;
```

Table 62: BCM RX Trap Flags

Name	***Purpose***
`BCM_RX_TRAP_UPDATE_ADD_VLAN`	Add VLAN to the final destination value
`BCM_RX_TRAP_UPDATE_FORWARDING_TYPE`	Replace the forwarding type

Table 63: Qualifier Forwarding Type (for bcm_forwarding_type)

Qualifier Forwarding Type	***Purpose:***
bcmForwardingTypeL2	L2 switching forwarding.
bcmForwardingTypeIp4Ucast	IPv4 Unicast Routing forwarding.
bcmForwardingTypeIp4Mcast	IPv4 Multicast Routing forwarding.
bcmForwardingTypeIp6Ucast	IPv6 Unicast Routing forwarding.
bcmForwardingTypeIp6Mcast	IPv6 Multicast Routing forwarding.
bcmForwardingTypeMpls	MPLS Switching forwarding.
bcmForwardingTypeTrill	Trill forwarding.
bcmForwardingTypeRxReason	Forwarding according to a RxReason.
bcmForwardingTypeTrafficManagement	Traffic Management forwarding, when an external Packet Processor sets the forwarding decision.
bcmForwardingTypeSnoop	Snooped packet.
bcmForwardingTypeFCoE	Fiber Channel over Ethernet forwarding.
bcmForwardingTypeCount	Always Last. Not a usable value.

Two new fields have been added for `bcm_rx_cfg_t` data structure for BCM RX APIs.

Table 64: bcm_rx_cfg_t Structure

Member	***Type***	***Description***
`num_of_cpu_addresses`	Explicitly set the relevant num of CPU addresses - ignore if not relevant.	
`cpu_address`	Explicitly set the relevant CPU addresses - ignore if not relevant.	

SERVICE ACTIVATION TEST (SAT)

The typo of field names in data type `bcm_sat_timestamp_format_t` have been corrected.

Table 65: SAT Timestamp Formats

Previous typo	*Corrected*
bcmSATimestampFormatIEEE1588v1	bcmSATTimestampFormatIEEE1588v1
bcmSATimestampFormatNTP	bcmSATTimestampFormatNTP

New SAT data types are defined and listed as following.

- `bcm_sat_gtf_t` is an identifier used to refer to a gtf object.
- `bcm_sat_ctf_t` is an identifier used to refer to a ctf object.
- `bcm_sat_header_type_t` is an enumeration of SAT Header Type.
- `bcm_sat_payload_type_t` is an enumeration of SAT Payload Type.
- `bcm_sat_stamp_type_t` is an enumeration of SAT Stamp Type.
- `bcm_sat_stamp_field_t` is an enumeration of SAT Stamp Field.
- `bcm_sat_gtf_rate_pattern_mode_t` is an enumeration of SAT gtf rate pattern mode.
- `bcm_sat_gtf_stat_counter_t` is an enumeration of SAT gtf counter types.
- `bcm_sat_event_type_t` is an enumeration of SAT event types for which callbacks can be registered.
- `bcm_sat_config_t` is a structure containing SAT common configuration.
- `bcm_sat_header_user_define_offsets_t` is a structure containing offsets of payload, timestamp and sequence number in user defined header.
- `bcm_sat_payload_t` is used to refer to a SAT payload object.
- `bcm_sat_stamp_t` is used to refer to a SAT stamp object.
- `bcm_sat_gtf_packet_edit_t` is a structure containing infomation about SAT gtf packet edit.
- `bcm_sat_gtf_packet_config_t` is a structure containing infomation about SAT gtf packet configuration, including header info, payload and packet edit info etc.
- `bcm_sat_gtf_bandwidth_t` is used to configure SAT gtf bandwidth.
- `bcm_sat_gtf_rate_pattern_t` is used to configure SAT gtf rate pattern.
- `bcm_sat_ctf_packet_info_t` is a structure containing infomation about SAT ctf packet configuration.
- `bcm_sat_ctf_identifier_t` is used to configure SAT ctf identifier.
- `bcm_sat_ctf_bin_limit_t` is an enumeration of SAT ctf bin limit.
- `bcm_sat_ctf_stat_config_t` is used to refer to a SAT ctf statistics configuration object.
- `bcm_sat_ctf_stat_t` is is a structure containing statistics infomation.
- `bcm_sat_ctf_availability_config_t` is used to configure parameters for ctf availability.
- `bcm_sat_ctf_report_config_t` is used to configure reports on ctf.
- `bcm_sat_report_event_data_t` is a structure containing report event data.

The data structures and defines of some above data types have been listed as following.

Table 66: SAT Header Type

Header Type	*Description*
bcmSatHeaderUserDefined	User defined SAT header
bcmSatHeaderY1731	SAT Header defined by Y1731
bcmSatHeaderMEF	SAT Header defined by MEF

Table 67: SAT Payload Type

Payload Type	Description
bcmSatPayloadConstant8Bytes	8 byte repeatable pattern
bcmSatPayloadConstant4Bytes	4 byte repeatable pattern
bcmSatPayloadPRBS	PRBS-31

Table 68: SAT Stamp Type

Stamp Type	Description
bcmSatStampInvalid	Invalid stamp type
bcmSatStampConstant2Bit	2 bit stamp - constant value to stamp into the packet
bcmSatStampCounter8Bit	8 bit counter
bcmSatStampCounter16Bit	16 bit counter

Table 69: SAT Stamp Field

Stamp Field	Description
bcmSatStampFieldUserDefined	Stamp user defined field
bcmSatStampFieldTc	Stamp Tc
bcmSatStampFieldDp	Stamp Dp
bcmSatStampFieldSrcMac	Stamp Src Mac
bcmSatStampFieldDestMac	Stamp Dest Mac
bcmSatStampFieldInnerVlan	Stamp Inner Vlan
bcmSatStampFieldOuterVlan	Stamp Outer Vlan

Table 70: SAT Rate Pattern Mode

Rate Pattern Mode	Description
bcmSatGtfRatePatternContinues	Burst only
bcmSatGtfRatePatternSimpleBurst	Interval only
bcmSatGtfRatePatternCombined	Combined mode

Table 71: SAT GTF Counter Types

GTF Counter Types	Description
bcmSatGtfStatPacketCount	Count number of packets transmitted

Table 72: SAT Event Types

SAT Event Types	Description
bcmSATEventReport	For Event Report

Table 73: SAT Miscellaneous defines

Name	Description
BCM_SAT_GTF_NUM_OF_PRIORITIES	Number of element in gtf.
BCM_SAT_PAYLOAD_MAX_PATTERN_SIZE	SAT payload pattern max size.
BCM_SAT_GTF_PACKET_LENGTH_NUM_OF_PATTERNS	Number of pattern of GTF packet length.
BCM_SAT_GTF_PACKET_MAX_PATTERN_LENGTH	SAT packet max pattern length.
BCM_SAT_GTF_MAX_STAMPS	Max stamps of SAT GTF.
BCM_SAT_CTF_IDENTIFIER_ANY	SAT ctf idenifier.
BCM_SAT_CTF_STAT_MAX_NUM_OF_BINS	SAT ctf bins number.

Table 74: SAT Config Structure Flag Definitions

Flag	Description
BCM_SAT_CONFIG_CRC_REVERSED_DATA	During CRC calculation the packet data bits are reversed
BCM_SAT_CONFIG_CRC_INVERT	During CRC calculation the result is inverted and reversed before stamping
BCM_SAT_CONFIG_CRC_HIGH_RESET	During CRC calculations, reset CRC value is taken to be 32'hffffffff
BCM_SAT_CONFIG_PRBS_USE_NXOR	During PRBS calculations the next bit is created using NXOR (and not XOR)

Table 75: SAT GTF Create Flag Definitions

Flag	Description
BCM_SAT_GTF_WITH_ID	Use the specified gtf ID

Table 76: SAT GTG Packet Config Structure Flag Definitions

Flag	Description
BCM_SAT_GTF_PACKET_EDIT_ADD_END_TLV	If set, add an END TLV at the end of the packet
BCM_SAT_GTF_PACKET_EDIT_ADD_CRC	If set, add CRC

Table 77: SAT GTF Rate Pattern Structure Flag Definitions

Flag	Description
BCM_SAT_GTF_RATE_PATTERN_STOP_INTERVAL_EQ_BURST	If set, INTERVAL state limit is set to the configuration, otherwise to the number of packets sent in BURST state

Table 78: SAT CTF Create Flag Definitions

Flag	Description
BCM_SAT_CTF_WITH_ID	Use the specified ctf ID

Table 79: SAT CTF Packet Configuration Structure Flag Definitions

Flag	Description
BCM_SAT_CTF_PACKET_INFO_ADD_END_TLV	If set, add end tlv, otherwise not add end tlv
BCM_SAT_CTF_PACKET_INFO_ADD_CRC	If set, add crc, otherwise not add crc

Table 80: SAT CTF Report Config Structure Flag Definitions

Flag	Description
BCM_SAT_CTF_REPORT_ADD_SEQ_NUM	If set, add sequence number to reports
BCM_SAT_CTF_REPORT_ADD_DELAY	If set, add delay to reports

```
    typedef struct bcm_sat_config_s {
        bcm_sat_timestamp_format_t timestamp_format; /* SAT timestamp format */
        uint32 config_flags;
    } bcm_sat_config_t;

    typedef struct bcm_sat_header_user_define_offsets_s {
        int payload_offset;     /* Payload offset */
        int timestamp_offset;   /* Timestamp offset */
        int seq_number_offset;  /* Sequence number offset */
    } bcm_sat_header_user_define_offsets_t;

    typedef struct bcm_sat_payload_s {
        bcm_sat_payload_type_t payload_type; /* Payload type */
        uint8 payload_pattern[BCM_SAT_PAYLOAD_MAX_PATTERN_SIZE]; /* Payload 
pattern */
    } bcm_sat_payload_t;

    typedef struct bcm_sat_stamp_s {
        bcm_sat_stamp_type_t stamp_type;     /* Stamp type */
        bcm_sat_stamp_field_t field_type;    /* Stamp field type */
        uint32 inc_step;                   /* Increment step - by what value the 
                           counter should be incremented */
        uint32 inc_period_packets;          /* Increment period - counter may be 
                           incremented every 1/2/4/8 packets */
        uint32 value;                        /* Stamp value */
        uint32 offset;                     /* Stamp offset in bytes - where to 
                           stamp the counter, beginning from 
                           start of packet */
    } bcm_sat_stamp_t;

    typedef struct bcm_sat_gtf_packet_edit_s {
        uint32 packet_length[BCM_SAT_GTF_PACKET_LENGTH_NUM_OF_PATTERNS]; /* 
Packet length */
        uint32 packet_length_pattern[BCM_SAT_GTF_PACKET_MAX_PATTERN_LENGTH]; /
* Packet length pattern */
        uint32 pattern_length;               /* Pattern length */
        bcm_sat_stamp_t stamps[BCM_SAT_GTF_MAX_STAMPS]; /* Configurae stamp */
       uint32 number_of_stamps;          /* Number of stamps valid in stamps */
```

```
        uint32 number_of_ctfs;              /* Number of CTFs which collect
                           information from the current gtf */
        uint32 flags;
    } bcm_sat_gtf_packet_edit_t;

    typedef struct bcm_sat_gtf_packet_config_s {
        bcm_sat_header_type_t sat_header_type; /* Sat header type */
        bcm_pkt_t header_info;              /* Sat header info */
        bcm_sat_payload_t payload;          /* Sat payload */
       bcm_sat_gtf_packet_edit_t packet_edit[BCM_SAT_GTF_NUM_OF_PRIORITIES]; /
* Packet edit for each element of the
                           gtf */
        uint32 packet_context_id;           /* Core ID for jericho */
        bcm_sat_header_user_define_offsets_t offsets; /* Offsets of payload,
timestamp and
                           sequence number in user defined
                           header */
    } bcm_sat_gtf_packet_config_t;

    typedef struct bcm_sat_gtf_bandwidth_s {
        uint32 rate;        /* traffic rate. Units: kbps */
        uint32 max_burst;   /* traffic burst. Units: kbit */
    } bcm_sat_gtf_bandwidth_t;

    typedef struct bcm_sat_gtf_rate_pattern_s {
       bcm_sat_gtf_rate_pattern_mode_t rate_pattern_mode; /* rate pattern mode
*/
        uint32 high_threshold;              /* High threshold */
        uint32 low_threshold;               /* Lower threshold */
        uint32 stop_iter_count;             /* Stop Iterations */
       uint32 stop_burst_count;            /* Stop Burst count - Burst state is
                           left when number of packets
                           transmitted in this state is
                           greater-equal than this value */
        uint32 stop_interval_count;         /* Stop interval count */
       uint32 burst_packet_weight;         /* Burst Pkt Weight - on each packet
                           transmitted in Burst state packet
                           counter increased by this value */
        uint32 interval_packet_weight;      /* Inteval Packet weight - on each
                           packet transmitted in Interval state
                           packet counter increased by this
                           value */
        uint32 flags;
    } bcm_sat_gtf_rate_pattern_t;

    typedef struct bcm_sat_ctf_packet_info_s {
        bcm_sat_header_type_t sat_header_type; /* Sat header type */
        bcm_sat_payload_t payload;          /* Sat payload */
        uint32 flags;
        bcm_sat_header_user_define_offsets_t offsets; /* Offsets of payload,
timestamp and
                           sequence number in user defined
                           header */
    } bcm_sat_ctf_packet_info_t;

    typedef struct bcm_sat_ctf_identifier_s {
        uint32 tc;          /* Class of service */
        uint32 color;       /* Color */
        uint32 trap_id;     /* Trap ID */
        uint32 session_id;  /* SAT session ID */
    } bcm_sat_ctf_identifier_t;

    typedef struct bcm_sat_ctf_bin_limit_s {
        uint32 bin_select;  /* bin select */
        uint32 bin_limit;   /* bin limit */
    } bcm_sat_ctf_bin_limit_t;

    typedef struct bcm_sat_ctf_stat_config_s {
        uint32 use_global_bin_config;       /* Use global bin config */
       uint32 bin_min_delay;               /* Bin min delay, for linear bins */
        uint32 bin_step;                    /* Bin step, for linear bins */
       uint32 update_counters_in_unvavail_state; /* Update counters in unavail
state */
    } bcm_sat_ctf_stat_config_t;

    typedef struct bcm_sat_ctf_stat_s {
        uint64 received_packet_cnt;         /* Counter packet counter */
       uint64 out_of_order_packet_cnt;     /* Counter for out of order packet */
        uint64 err_packet_cnt;              /* Counter for error packet */
        uint64 received_octet_cnt;          /* Counter for received octet */
       uint64 out_of_order_octet_cnt;      /* Counter for out of order octet */
        uint64 err_octet_cnt;               /* Counter for error octet */
        uint64 last_packet_delay;          /* counter for last packet delay */
      uint64 bin_delay_counters[BCM_SAT_CTF_STAT_MAX_NUM_OF_BINS]; /* Counters
for bin delay */
       uint64 accumulated_delay_count;     /* Counter for accumulated delay */
        uint64 max_packet_delay;            /* max packet delay */
        uint64 min_packet_delay;            /* min packet delay */
        uint64 time_of_day_stamp;           /* Time of day stamp */
       uint64 sec_in_avail_state_counter;  /* Counter for sec in avail state */
    } bcm_sat_ctf_stat_t;

    typedef struct bcm_sat_ctf_availability_config_s {
        uint32 switch_state_num_of_slots;   /* Number of slot */
        uint32 switch_state_threshold_per_slot; /* Threshold per slot */
    } bcm_sat_ctf_availability_config_t;

    typedef struct bcm_sat_ctf_report_config_s {
        uint32 report_sampling_percent; /* Report sampling percent */
        uint32 flags;
    } bcm_sat_ctf_report_config_t;

    typedef struct bcm_sat_report_event_data_s {
        int sequence_number;        /* Sequence number */
        int sequence_number_valid;  /* Sequence number valid */
        int delay;                  /* Delay */
        int delay_valid;            /* Delay valid */
    } bcm_sat_report_event_data_t;
```

New SAT APIs have been added as following.

bcm_sat_header_user_define_offsets_t_init

Initialize a bcm_sat_header_user_define_offsets_t structure.

Syntax

```
#include <bcm/sat.h>
void
bcm_sat_header_user_define_offsets_t_init(bcm_sat_header_user_define_offsets_t
*offsets);
```

Parameters

`offsets`	(OUT) Pointer toSAT user define offsets structure to initialize

Description

Initializes a SAT user define offsets structure to default values.

bcm_sat_payload_t_init

Initialize a bcm_sat_payload_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_payload_t_init(bcm_sat_payload_t *payload);
```

Parameters

`payload`	(OUT) Pointer to SAT payload structure to initialize

Description

Initializes a SAT payload structure to default values.

bcm_sat_stamp_t_init

Initialize a bcm_sat_stamp_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_stamp_t_init(bcm_sat_stamp_t *stamp);
```

Parameters

`stamp`	(OUT) Pointer to SAT stamp structure to initialize

Description

Initializes a SAT stamp structure to default values.

bcm_sat_ctf_availability_config_t_init

Initialize a bcm_sat_ctf_availability_config_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_ctf_availability_config_t_init(bcm_sat_ctf_availability_config_t
*config);
```

Parameters

`config`	(OUT) Pointer to SAT ctf availability config structure to initialize

Description

Initializes a SAT ctf availability config structure to default values. This function must be used to initialize any SAT ctf availability config structure before passing it to an API

bcm_sat_ctf_availability_config_set

Set ctf availability.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_availability_config_set(
    int unit,
    bcm_sat_ctf_t ctf_id,
    bcm_sat_ctf_availability_config_t *config);
```

Parameters

`unit`	(IN) BCM device number
`ctf_id`	(IN) CTF ID
`config`	(IN) Pointer to SAT availability config structure

Description

Set ctf availability.

Returns

bcm_sat_config_t_init

Initialize a bcm_sat_config_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_config_t_init(bcm_sat_config_t *conf);
```

Parameters

conf	(OUT) Pointer to SAT config structure to initialize

Description

Initializes a SAT config structure to default values.

bcm_sat_config_set

Set common configuration.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_config_set(
    int unit,
    bcm_sat_config_t *conf);
```

Parameters

unit	(IN) BCM device number
conf	(IN) Pointer to SAT config structure

Description

Set common configuration.

Returns

bcm_sat_config_get

Get common configuration.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_config_get(
    int unit,
    bcm_sat_config_t *conf)
```

Parameters

`unit`	(IN) BCM device number
`conf`	(IN) Pointer to SAT config structure

Description

Get common configuration.

Returns

bcm_sat_gtf_create

Create SAT GTF object.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_gtf_create(
    int unit,
    uint32 flags,
    bcm_sat_gtf_t *gtf_id);
```

Parameters

`unit`	(IN) BCM device number
`flags`	(IN) `BCM_SAT_GTF_WITH_ID: gtf_id` argument is given.
`gtf_id`	(IN/OUT) GTF ID

Description

If `BCM_SAT_GTF_WITH_ID` is not set, an ID is created for the GTF. Otherwise, the specified ID is used.

Returns

bcm_sat_gtf_destroy

Destory a SAT GTF object.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_gtf_destroy(
    int unit,
    bcm_sat_gtf_t gtf_id);
```

Parameters

`unit`	(IN) BCM device number
`gtf_id`	(IN) GTF ID

Description

Destory a SAT GTF object and disable generate Packet.

Returns

bcm_sat_gtf_destroy_all

Destory all SAT GTF objects.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_gtf_destroy_all(
    int unit);
```

Parameters

`unit`	(IN) BCM device number

Description

Destory all SAT GTF objects.

Returns

bcm_sat_gtf_traverse

Traverse all the gtfs.

Syntax

```
        typedef int (*bcm_sat_gtf_traverse_cb)(
            int unit,
            bcm_sat_gtf_t gtf_id,
            void *user_data);

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

```
int bcm_sat_gtf_traverse(
    int unit,
    bcm_sat_gtf_traverse_cb cb,
    void *user_data);
```

Parameters

`unit`	(IN) BCM device number
`cb`	(IN) Pointer to the callback function
`user_data`	(OUT) Pointer to arbitrary user data to be passed back to the callback

Description

Calls the specified callback for each gtf.

Returns

bcm_sat_gtf_packet_config_t_init

Initialize a bcm_sat_gtf_packet_config_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_gtf_packet_config_t_init(bcm_sat_gtf_packet_config_t *pkt_cfg);
```

Parameters

`pkt_cfg`	(OUT) Pointer to SAT packet config structure to initialize

Description

Initializes a SAT gtf packet config structure to default values. This function must be used to initialize any SAT gtf packet config structure before passing it to an API function.

bcm_sat_gtf_packet_config

Set gtf packet format.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_gtf_packet_config(
    int unit,
    bcm_sat_gtf_t gtf_id,
    bcm_sat_gtf_packet_config_t *config);
```

Parameters

`unit`	(IN) BCM device number
`gtf_id`	(IN) GTF ID
`config`	(IN) Pointer to SAT GTF Packet config structure

Description

Set packet format for gtf.

Returns

bcm_sat_gtf_bandwidth_t_init

Initialize a bcm_sat_gtf_bandwidth_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_gtf_bandwidth_t_init(bcm_sat_gtf_bandwidth_t *bw);
```

Parameters

`bw`	(OUT) Pointer to SAT gtf bandwidth structure to initialize

Description

Initializes a SAT gtf bandwidth structure to default values. This function must be used to initialize any SAT gtf bandwidth structure before passing it to an API function.

bcm_sat_gtf_bandwidth_set

Set gtf bandwidth.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_gtf_bandwidth_set(
    int unit,
    bcm_sat_gtf_t gtf_id,
    int priority,
    bcm_sat_gtf_bandwidth_t *bw);
```

Parameters

`unit`	(IN) BCM device number
`gtf_id`	(IN) GTF ID
`priority`	(IN) -1 indicates common; 0 indicates cir; 1 indicates pir
`bw`	(IN) Pointer to SAT bandwidth structure

Description

Set gtf bandwidth.

Returns

bcm_sat_gtf_bandwidth_get

Get gtf bandwidth.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_gtf_bandwidth_get(
    int unit,
    bcm_sat_gtf_t gtf_id,
    int priority,
    bcm_sat_gtf_bandwidth_t *bw);
```

Parameters

`unit`	(IN) BCM device number
`gtf_id`	(IN) GTF ID
`priority`	(IN) -1 indicates common; 0 indicates cir; 1 indicates pir
`bw`	(IN/OUT) Pointer to SAT bandwidth structure

Description

Get gtf bandwidth.

Returns

bcm_sat_gtf_rate_pattern_t_init

Initialize a bcm_sat_gtf_rate_pattern_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_gtf_rate_pattern_t_init(bcm_sat_gtf_rate_pattern_t
*rate_pattern);
```

Parameters

`rate_pattern`	(OUT) Pointer to SAT gtf rate pattern structure to initialize

Description

Initializes a SAT gtf rate pattern structure to default values. This function must be used to initialize any SAT gtf rate pattern structure before passing it to an API function.

bcm_sat_gtf_rate_pattern_set

Set gtf rate pattern.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_gtf_rate_pattern_set(
    int unit,
    bcm_sat_gtf_t gtf_id,
    int priority,
    bcm_sat_gtf_rate_pattern_t *config);
```

Parameters

`unit`	(IN) BCM device number
`gtf_id`	(IN) GTF ID
`priority`	(IN) -1 indicates common; 0 indicates cir; 1 indicates pir
`config`	(IN) Pointer to SAT rate pattern structure

Description

Set gtf rate pattern.

Returns

bcm_sat_gtf_rate_pattern_get

Get gtf rate pattern.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_gtf_rate_pattern_get(
    int unit,
    bcm_sat_gtf_t gtf_id,
    int priority,
    bcm_sat_gtf_rate_pattern_t *config);
```

Parameters

`unit`	(IN) BCM device number
`gtf_id`	(IN) GTF ID
`priority`	(IN) -1 indicates common; 0 indicates cir; 1 indicates pir
`config`	(IN/OUT) Pointer to SAT rate pattern structure

Description

Get gtf rate pattern.

Returns

bcm_sat_gtf_stat_get

Get gtf Statistics.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_gtf_stat_get(
    int unit,
    bcm_sat_gtf_t gtf_id,
    int priority,
    uint32 flags,
    bcm_sat_gtf_stat_counter_t type,
    uint64 *value);
```

Parameters

`unit`	(IN) BCM device number
`gtf_id`	(IN) GTF ID
`priority`	(IN) -1 indicates common; 0 indicates cir; 1 indicates pir
`flags`	(IN) flags
`type`	(IN) stat counter type
`value`	(OUT) Pointer counter value

Description

Get gtf Statistics.

Returns

bcm_sat_ctf_create

Create SAT CTF object.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_create(
    int unit,
    uint32 flags,
    bcm_sat_ctf_t *ctf_id);
```

Parameters

`unit`	(IN) BCM device number
`flags`	(IN) `BCM_SAT_CTF_WITH_ID: ctf_id` argument is given.
`ctf_id`	(IN/OUT) CTF ID

Description

If `BCM_SAT_CTF_WITH_ID` is not set, an ID is created for the CTF. Otherwise, the specified ID is used.

Returns

bcm_sat_ctf_destroy

Destory a SAT CTF object.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_destroy(
    int unit,
    bcm_sat_ctf_t ctf_id);
```

Parameters

`unit`	(IN) BCM device number
`gtf_id`	(IN) CTF ID

Description

Destory a SAT CTF object and disable generate Packet.

Returns

bcm_sat_ctf_destroy_all

Destory all SAT CTF objects.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_destroy_all(
    int unit);
```

Parameters

`unit`	(IN) BCM device number

Description

Destory all SAT CTF objects.

Returns

bcm_sat_ctf_traverse

Traverse all the ctfs.

Syntax

```
typedef int (*bcm_sat_ctf_traverse_cb)(
    int unit,
    bcm_sat_ctf_t ctf_id,
    void *user_data);

#include <bcm/sat.h>
int bcm_sat_ctf_traverse(
    int unit,
    bcm_sat_ctf_traverse_cb cb,
    void *user_data);
```

Parameters

unit	(IN) BCM device number
cb	(IN) Pointer to the callback function
user_data	(OUT) Pointer to arbitrary user data to be passed back to the callback

Description

Calls the specified callback for each ctf.

Returns

bcm_sat_ctf_packet_info_t_init

Initialize a bcm_sat_ctf_packet_info_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_ctf_packet_info_t_init(bcm_sat_ctf_packet_info_t *packet_info);
```

Parameters

`packet_info`	(OUT) Pointer to SAT ctf packet info structure to initialize

Description

Initializes a SAT ctf packet info structure to default values. This function must be used to initialize any SAT ctf packet info structure before passing it to an API function.

bcm_sat_ctf_packet_config

Set ctf packet format.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_packet_config(
    int unit,
    bcm_sat_ctf_t ctf_id,
    bcm_sat_ctf_packet_info_t *packet_info);
```

Parameters

`unit`	(IN) BCM device number
`ctf_id`	(IN) CTF ID
`packet_info`	(IN) Pointer to SAT CTF Packet config structure

Description

Set packet format for gtf.

Returns

bcm_sat_ctf_identifier_t_init

Initialize a bcm_sat_ctf_identifier_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_ctf_identifier_t_init(bcm_sat_ctf_identifier_t *ctf_identifier);
```

Parameters

`ctf_identifier`	(OUT) Pointer to SAT ctf identifier structure to initialize

Description

Initializes a SAT ctf identifier structure to default values. This function must be used to initialize any SAT ctf identifier structure before passing it to an API function.

bcm_sat_ctf_identifier_map

Map identifier to ctf.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_identifier_map(
    int unit,
    bcm_sat_ctf_identifier_t *identifier,
    bcm_sat_ctf_t ctf_id);
```

Parameters

`unit`	(IN) BCM device number
`identifier`	(IN) Pointer to SAT Identifier structure
`ctf_id`	(IN) CTF ID

Description

Map identifier to ctf.

Returns

bcm_sat_ctf_identifier_unmap

Unmap identifier to ctf.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_identifier_unmap(
    int unit,
    bcm_sat_ctf_identifier_t *identifier);
```

Parameters

`unit`	(IN) BCM device number
`identifier`	(IN) Pointer to SAT Identifier structure

Description

Unmap identifier to ctf.

Returns

bcm_sat_ctf_trap_add

Add specific trap ID.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_trap_add(
    int unit,
    uint32 trap_id);
```

Parameters

`unit`	(IN) BCM device number
`trap_id`	(IN) Trap ID

Description

Add specific trap ID

Returns

bcm_sat_ctf_trap_remove

Remove specific trap ID.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_trap_remove(
    int unit,
    uint32 trap_id);
```

Parameters

`unit`	(IN) BCM device number
`trap_id`	(IN) Trap ID

Description

Remove specific trap ID.

Returns

bcm_sat_ctf_trap_remove_all

Destory all Trap objects.

Syntax

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

```
int bcm_sat_ctf_trap_remove_all(
    int unit);
```

Parameters

`unit`	(IN) BCM device number

Description

Destory all Trap objects.

Returns

bcm_sat_ctf_bin_limit_t_init

Initialize a bcm_sat_ctf_bin_limit_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_ctf_bin_limit_t_init(bcm_sat_ctf_bin_limit_t *bins);
```

Parameters

`bins`	(OUT) Pointer to SAT ctf bin limit structure to initialize

Description

Initializes a SAT ctf bin limit structure to default values. This function must be used to initialize any SAT ctf bin limit structure before passing it to an API function.

bcm_sat_ctf_bin_limit_set

Set ctf bin limit configuration.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_bin_limit_set(
    int unit,
    int bins_count,
    bcm_sat_ctf_bin_limit_t *bins);
```

Parameters

`unit`	(IN) BCM device number
`bins_count`	(IN) Number of elements in array parameter
`bins`	(IN) Pointer to SAT bin limit structure array

Description

Set ctf bin limit configuration.

Returns

bcm_sat_ctf_bin_limit_get

Get ctf bin limit configuration.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_bin_limit_get(
    int unit,
    int max_bins_count,
    int *bins_count,
    bcm_sat_ctf_bin_limit_t *bins);
```

Parameters

`unit`	(IN) BCM device number
`max_bins_count`	(IN) Number of elements in array parameter
`bins_count`	(OUT) Number of bins retrieved
`bins`	(OUT) Pointer to SAT bin limit structure array

Description

Get ctf bin limit configuration.

Returns

bcm_sat_ctf_stat_config_t_init

Initialize a bcm_sat_ctf_stat_config_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_ctf_stat_config_t_init(bcm_sat_ctf_stat_config_t *stat_cfg);
```

Parameters

`stat_cfg`	(OUT) Pointer to SAT ctf stat config structure to initialize

Description

Initializes a SAT ctf stat config structure to default values. This function must be used to initialize any SAT ctf stat config structure before passing it to an API function.

bcm_sat_ctf_stat_config_set

Configure ctf stat.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_stat_config_set(
    int unit,
    bcm_sat_ctf_t ctf_id,
    bcm_sat_ctf_stat_config_t *stat);
```

Parameters

`unit`	(IN) BCM device number
`stat`	(IN) Pointer to SAT ctf stat config structure
`ctf_id`	(IN) CTF ID

Description

Configure ctf stat.

Returns

bcm_sat_ctf_stat_t_init

Initialize a bcm_sat_ctf_stat_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_ctf_stat_t_init(bcm_sat_ctf_stat_t *stat);
```

Parameters

`stat`	(OUT) Pointer to SAT ctf stat structure to initialize

Description

Initializes a SAT ctf stat structure to default values. This function must be used to initialize any SAT ctf stat structure before passing it to an API function.

bcm_sat_ctf_stat_get

Get ctf stat values.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_stat_get(
    int unit,
    bcm_sat_ctf_t ctf_id,
    uint32 flags,
    bcm_sat_ctf_stat_t *stat);
```

Parameters

`unit`	(IN) BCM device number
`flags`	(IN) `BCM_SAT_CTF_STAT_DO_NOT_CLR_ON_READ:` do not clear counter when reading
`ctf_id`	(IN) CTF ID
`stat`	(IN) Pointer to SAT ctf stat structure

Description

Get ctf stat values.

Returns

bcm_sat_ctf_trap_data_to_session_map

Map trap data to session ID.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_trap_data_to_session_map(
    int unit,
    uint32 trap_data,
    uint32 trap_data_mask,
    uint32 session_id);
```

Parameters

`unit`	(IN) BCM device number
`trap_data`	(IN) trap data
`trap_data_mask`	(IN) trap data mask
`session_id`	(IN) session id

Description

Map trap data to session ID.

Returns

bcm_sat_ctf_trap_data_to_session_unmap

Unmap trap data to session ID.

Syntax

```
#include <bcm/sat.h>
extern int bcm_sat_ctf_trap_data_to_session_unmap(
    int unit,
    uint32 trap_data,
    uint32 trap_data_mask);
```

Parameters

`unit`	(IN) BCM device number
`trap_data`	(IN) trap data
`trap_data_mask`	(IN) trap data mask

Description

Unmap trap data to session ID.

Returns

bcm_sat_ctf_report_config_t_init

Initialize a bcm_sat_ctf_report_config_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_ctf_report_config_t_init(bcm_sat_ctf_report_config_t *reports);
```

Parameters

`reports`	(OUT) Pointer to SAT ctf report config structure to initialize

Description

Initializes a SAT ctf report config structure to default values. This function must be used to initialize any SAT ctf report config structure before passing it to an API function.

bcm_sat_ctf_reports_config_set

Set ctf report configuration.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_ctf_reports_config_set(
    int unit,
    bcm_sat_ctf_t ctf_id,
    bcm_sat_ctf_report_config_t *reports);
```

Parameters

`unit`	(IN) BCM device number
`ctf_id`	(IN) CTF ID
`reports`	(IN) Pointer to SAT report config structure

Description

Set ctf report configuration.

Returns

bcm_sat_event_register

Register a handler for SAT events.

Syntax

```
typedef int (*bcm_sat_event_cb)(
    int unit,
    bcm_sat_event_type_t event_type,
    void *user_provided_struct,
    void *user_data);

#include <bcm/sat.h>
int bcm_sat_event_register(int unit, bcm_sat_event_types_t event_type,
bcm_sat_event_cb cb, void *user_data);
```

Parameters

`unit`	BCM device number
`event_type`	(IN) A collection of event types for which the specified callback will be called.
`cb`	(IN) Pointer to the callback function
`user_data`	(OUT) Pointer to arbitrary user data to be passed back to the callback

Description

Registers the specified callback to handle the event types specified in the collection.

Returns

bcm_sat_event_unregister

Unregister a handler for SAT events.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_event_unregister(int unit, bcm_sat_event_types_t event_type,
bcm_sat_event_cb cb);
```

Parameters

`unit` BCM device number

`event_type` (IN) A collection of event types for which the specified callback will no longer be called.

`cb` (IN) Pointer to the callback function

Description

Unregisters the specified callback from handling the event types specified in the collection. The callback will continue to be called for any other event types for which it remains registered.

Returns

SWITCH CONTROL

Below new Switch control types have been added in this release.

Table 81: Switch Type Values

Value	*Description*	*Arg Value*
`bcmSwitchHashL2MPLSPayloadSelect0`	Set hash control to select L2 MPLS payload L2/L3 fields for Hash Block A.	• `BCM_HASH_SELECT_INNER_L2` : Inner L2 fields • `BCM_HASH_SELECT_INNER_L3` : Inner L3 fields
`bcmSwitchHashL2MPLSPayloadSelect1`	Set hash control to select L2 MPLS payload L2/L3 fields for Hash Block B.	• `BCM_HASH_SELECT_INNER_L2` : Inner L2 fields • `BCM_HASH_SELECT_INNER_L3` : Inner L3 fields

Table 81: Switch Type Values

Value	Description	Arg Value
bcmSwitchHashL3L2MPLSField0	BCM56340/BCM56450/BCM56640/ BCM56850 Hash A Selection of L3 Payload for L2 MPLS Packets.	• BCM_HASH_FIELD_DSTMOD - Destination module ID • BCM_HASH_FIELD_DSTPORT - Destination port ID • BCM_HASH_FIELD_SRCMOD - Source module ID • BCM_HASH_FIELD_SRCPORT - Source port ID • BCM_HASH_FIELD_PROTOCOL Protocol ID • BCM_HASH_FIELD_DSTL4 - Destination L4 port • BCM_HASH_FIELD_SRCL4 - Source L4 port • BCM_HASH_FIELD_VLAN - VLAN ID • BCM_HASH_FIELD_IP4DST_LO - IPv4 destination address lower 16 bits • BCM_HASH_FIELD_IP4DST_HI - IPv4 destination address upper 16 bits • BCM_HASH_FIELD_IP4SRC_LO - IPv4 source address lower 16 bits • BCM_HASH_FIELD_IP4SRC_HI - IPv4 source address upper 16 bits • BCM_HASH_FIELD_CNTAG_FLOW_ID - Congestion notification tag flow identifier (BCM56840 only)

Table 81: Switch Type Values

Value	Description	Arg Value
bcmSwitchHashL3L2MPLSField1	BCM56340/BCM56450/BCM56640/ BCM56850 Hash B Selection of L3 Payload for L2 MPLS Packets.	• BCM_HASH_FIELD_DSTMOD - Destination module ID • BCM_HASH_FIELD_DSTPORT - Destination port ID • BCM_HASH_FIELD_SRCMOD - Source module ID • BCM_HASH_FIELD_SRCPORT - Source port ID • BCM_HASH_FIELD_PROTOCOL Protocol ID • BCM_HASH_FIELD_DSTL4 - Destination L4 port • BCM_HASH_FIELD_SRCL4 - Source L4 port • BCM_HASH_FIELD_VLAN - VLAN ID • BCM_HASH_FIELD_IP4DST_LO - IPv4 destination address lower 16 bits • BCM_HASH_FIELD_IP4DST_HI - IPv4 destination address upper 16 bits • BCM_HASH_FIELD_IP4SRC_LO - IPv4 source address lower 16 bits • BCM_HASH_FIELD_IP4SRC_HI - IPv4 source address upper 16 bits • BCM_HASH_FIELD_CNTAG_FLOW_ID - Congestion notification tag flow identifier (BCM56840 only)
cw bcmSwitchHashIPSecSelect0	Set hash control to select outer L4_SRC/DST_PORT or IPV6 SPI for hash block A.	• BCM_HASH_IPSEC_SELECT_OUTER_L4_PORT • BCM_HASH_IPSEC_SELECT_SPI
cw bcmSwitchHashIPSecSelect1	Set hash control to select outer L4_SRC/DST_PORT or IPV6 SPI for hash block B.	• BCM_HASH_IPSEC_SELECT_OUTER_L4_PORT • BCM_HASH_IPSEC_SELECT_SPI

Table 81: Switch Type Values

Value	Description	Arg Value
`bcmSwitchAlternateStoreForward`	Configure faster packet store and forward mechanism. Also known as cut-through forwarding.	For BCM56960: • bcmPortAsfModeStoreAndForward[0] Store and Forward (SAF) mode • bcmPortAsfModeSameSpeed [1] Cut-thru forwarding between same speed ports • bcmPortAsfModeSlowToFast [2] Cut-thru forwarding between slower to faster speed ports • bcmPortAsfModeFastToSlow [3] Cut-thru forwarding between faster to slower speed ports BCM56960 supports aforementioned additional modes of cut-through.
`bcmSwitchECMPLevel1RandomSeed`	Seed used by Random Number Generator in Randomized Load Balancing for ECMP(at Level 1 in Hierarchical ECMP Mode).	Range of values: 0x1 - 0xFFFF
`bcmSwitchECMPLevel2RandomSeed`	Seed used by Random Number Generator in Randomized Load Balancing for ECMP at Level 2 in Hierarchical ECMP Mode. Not used in Single Level Mode.	Range of values: 0x1 - 0xFFFF
`bcmSwitchTimesyncEgressTimestampingMode`	Selects 32bit or 48bit timestamping mode for 1588 packets.	• bcmTimesyncTimestampingMode32bit - 32bit timestamping mode • bcmTimesyncTimestampingMode48bit - 48bit timestamping mode
`bcmSwitchSynchronousPortClockSourceValid`	Returns whether synchronous clock can be sucessfully recovered from the port, configured via bcmSwitchSynchronousPortClockSource	
`bcmSwitchSynchronousPortClockSourceBkupValid`	Returns whether synchronous clock is successfully recovered from the back up port, configured via bcmSwitchSynchronousPortClockSourceBkup	

The flag `BCM_SWITCH_PKT_INFO_HASH_LBID` is now added into the `flags` field in the Switch packet information structure.

Table 82: BCM Switch packet information flags

Name	Purpose
`BCM_SWITCH_PKT_INFO_HASH_LBID`	Retrieve LBID.

New switch APIs are added as following.

bcm_switch_hash_entry_config_t_init

Initialize a bcm_switch_hash_entry_config_t structure.

Syntax

```
#include <bcm/switch.h>
void
bcm_switch_hash_entry_config_t_init(bcm_switch_hash_entry_config_t *config);
```

Parameters

`config`	(IN/OUT) flex hashing configuration parameters

Description

Initializes a flex hashing configuration parameters structure to default values. This function should be used to initialize any such 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_switch_hash_entry_config_t` structure, and `bcm_switch_hash_entry_config_t_init` will initialize the new members to correct default values.

Returns

Nothing

bcm_switch_hash_entry_set

Set a flex hash entry configuration information.

Syntax

```
#include <bcm/switch.h>
int bcm_switch_hash_entry_set( int unit, bcm_hash_entry_t entry,
bcm_switch_hash_entry_config_t *config);
```

Parameters

`unit`	(IN) BCM device number
`entry`	(IN) BCM hash entry
`config`	(IN) BCM hash configuration information

Description

This API is used to set the flex hash configuration information.

Returns

bcm_switch_hash_entry_get

Get a flex hash entry configuration information.

Syntax

```
#include <bcm/switch.h>
int bcm_switch_hash_entry_get( int unit, bcm_hash_entry_t entry,
bcm_switch_hash_entry_config_t *config);
```

Parameters

`unit`	(IN) BCM device number
`entry`	(IN) BCM hash entry
`config`	(IN/OUT) BCM hash configuration information

Description

This API is used to get the flex hash configuration information.

Returns

- `bcm_switch_hash_field_t` enumerates switch hash fields supported by the API shown in `Switch hash fields (`page 97`)`

Table 83: Switch hash fields

Name	***Purpose***
bcmSwitchFlexHashField0 = 0,	field 0 in flexible hash RAM table.
bcmSwitchFlexHashField1 = 1,	field 1 in flexible hash RAM table.
bcmSwitchFlexHashFieldCount = 2	Number of fields in this enum.

- The following macro defines a specific offset value to indicate that using `bcm_switch_hash_entry_config_t` structure to setup the offset and mask instead of using `bcm_switch_hash_entry_install` directly.

```
    #define BCM_SWITCH_HASH_USE_CONFIG  (~0U)
```

- The following macro defines mask array length in `bcm_switch_hash_entry_config_t` struct.

```
    #define BCM_SWITCH_HASH_ENTRY_CONFIG_MASK_LENGTH 4
```

- The following defines user-supplied callback functions for the switch hash traversal API.

```
    typedef int (*bcm_switch_hash_entry_traverse_cb)(int unit, int flags,
bcm_hash_entry_t entry, void *user_data);
```

- `bcm_switch_hash_entry_config_t` contains information used when setting or retrieving flex hash entry. The description for each API below describes which structure members are used for inputs and outputs.
- Always initialize an flex hash configuration structure using the `bcm_switch_hash_entry_config_t_init` function prior to passing it to an API, because future API releases may add more structure members.

```
    typedef struct bcm_switch_hash_entry_config_s {
        uint32 flags;                        /* flags should be zero for today
*/
         bcm_switch_hash_field_t field;       /* field selector, currently
there are
                                   two fields in flex hash ram table */
        uint32 offset;                        /* hash offset */
        int mask_length;                     /* number of bytes in the mask */
         uint8 mask[BCM_SWITCH_HASH_ENTRY_CONFIG_MASK_LENGTH]; /* hash mask
*/
        bcm_udf_layer_t base_layer;          /* this is unsupported for now */
    } bcm_switch_hash_entry_config_t;
```

bcm_switch_hash_entry_traverse

Traverse all configured flex hash entry to callback.

Syntax

```
#include <bcm/switch.h>
int bcm_switch_hash_entry_traverse( int unit, int flags,
bcm_switch_hash_entry_traverse_cb cb_fn, void *user_data);
```

Parameters

`unit`	(IN) BCM device number
`flags`	(IN) BCM hash flags
`cb_fn`	(IN) BCM hash callback function
`user_data`	(IN) User data to be passed to callback function

Description

This API is used to traverse the flex hash entries.

Returns

bcm_switch_hash_qualify_udf

Set a flex hash entry configuration information.

Syntax

```
#include <bcm/switch.h>
int bcm_switch_hash_qualify_udf( int unit, bcm_hash_entry_t entry, bcm_udf_id_t
udf_id, int length, uint8 *data, uint8 *mask);
```

Parameters

`unit`	(IN) BCM device number
`entry`	(IN) BCM hash entry
`udf_id`	(IN) BCM udf identifier
`length`	(IN) Qualifier length in BYTE
`data`	(IN) Qualifier data
`mask`	(IN) Qualifier mask

Description

This API is used to configure the flex hash qualifier data and mask.

Returns

bcm_switch_hash_qualify_udf_get

Get a flex hash entry configuration information.

Syntax

```
#include <bcm/switch.h>
int bcm_switch_hash_qualify_udf_get( int unit, bcm_hash_entry_t entry,
bcm_udf_id_t udf_id, int max_length, uint8 *data, uint8 *mask, int
*actual_length);
```

Parameters

`unit`	(IN) BCM device number
`entry`	(IN) BCM hash entry
`udf_id`	(IN) BCM udf identifier
`max_length`	(IN) Max qualifier length in BYTE
`data`	(OUT) Qualifier data
`mask`	(OUT) Qualifier mask
`actual_length`	(OUT) Actual qualifier length in BYTE

Description

This API is used to get the flex hash qualifier configuration information.

Returns

bcm_switch_tpid_add

Add a global TPID.

Syntax

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

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 */
} bcm_switch_tpid_type_t;

typedef struct bcm_switch_tpid_info_s {
    bcm_switch_tpid_type_t tpid_type;   /* The type of TPID to be set */
    uint16 tpid_value;                  /* The value of TPID to be set */
    uint32 flags;                     /* Future expansion, BCM_SWITCH_TPID_XXX
   flags */
    int color;                          /* Color mode for TPID */
} bcm_switch_tpid_info_t;

int
bcm_switch_tpid_add(
    int unit,
    uint32 options,
    bcm_switch_tpid_info_t *tpid_info);
```

Parameters

`unit`	(IN) Unit number.
`options`	(IN) Future expansion i.e, REPLACE option.
`tpid_info`	(IN) TPID info includes key: `{tpid_type, tpid_value}` and Associated value(s): {flags, color}.

Description

This API is used to add a global TPID. Global TPIDs are used as a supply for ingress parser to get tag format of incoming packets, besides TPIDs configured per port.

Returns

`BCM_E_xxx`

bcm_switch_tpid_delete

Delete a global TPID.

Syntax

```
#include <bcm/switch.h>
int
bcm_switch_tpid_delete(
    int unit,
    bcm_switch_tpid_info_t *tpid_info);
```

Parameters

`unit`	(IN) Unit number.
`tpid_info`	(IN) TPID info includes key: {`tpid_type, tpid_value`} and Associated value(s): {flags, color}.

Description

This API is used to delete a global TPID.

Returns

`BCM_E_xxx`

bcm_switch_tpid_delete_all

Delete all global TPIDs.

Syntax

```
#include <bcm/switch.h>
int
bcm_switch_tpid_delete_all(
    int unit);
```

Parameters

`unit`	(IN) Unit number.

Description

This API is used to delete all global TPIDs.

Returns

`BCM_E_xxx`

bcm_switch_tpid_get

Get a global TPID.

Syntax

```
#include <bcm/switch.h>
int
bcm_switch_tpid_get(
    int unit,
    bcm_switch_tpid_info_t *tpid_info);
```

Parameters

unit	(IN) Unit number.
tpid_info	(OUT) TPID info includes key: {tpid_type, tpid_value} and Associated value(s): {flags, color}.

Description

This API is used to retrieve a global TPID.

Returns

BCM_E_xxx

bcm_switch_tpid_get_all

Get global TPID count or TPIDs.

Syntax

```
#include <bcm/switch.h>
int
bcm_switch_tpid_get_all(
    int unit,
    int size,
    bcm_switch_tpid_info_t *tpid_info_array,
    int *count);
```

Parameters

unit	(IN) Unit number.
size	(IN) Size of tpid_info_array to be got.
tpid_info_array	(OUT) TPID info array.
count	(OUT) Item count of tpid_info_array.

Description

This API is used to retrieve the count of global TPIDs configured in system or mutiple global TPIDs based on size.

Returns

BCM_E_xxx

bcm_switch_pkt_trace_info_get

Obtain switch results of packet inserted on the given port
(currently disabled)

Syntax

```
#include <bcm/switch.h>
int bcm_esw_switch_pkt_trace_info_get(
 int    unit,
 uint32 options,
 uint8  port,
 int    len,
 uint8 *data,
 bcm_switch_pkt_trace_info_t *pkt_trace_info);
```

Parameters

`unit`	(IN) Unit number.
`options`	(IN) `cpu_pkt_profile` register options
`len`	(IN) Number of bytes to copy from data
`data`	(IN) Source packet buffer to copy from
`pkt_trace_src_port`	(IN) a local front panel port that acts as a masquerade port
`pkt_trace_info_result`	(INOUT) visibility packet process information in `bcm_pkt_trace_info_s` format

Description

This API is used to retrieve the switching information of packet inserted on the give port for `options,BCM_PKT_TRACE_LEARN, BCM_PKT_TRACE_NO_IFP,` and `BCM_PKT_TRACE_FORWARD` can be set to control trace packet behavior (currently disabled)

```
typedef struct bcm_switch_pkt_trace_info_s {
     uint32 ivp_raw_data[4];      /* packet trace ivp process result data */
    uint32 isw1_raw_data[8];     /* packet trace isw1 process result data */
    uint32 isw2_raw_data[16];    /* packet trace isw2 process result data */
    uint8 dest_pipe_num;         /* ingress pipeline number of the pkt trace
                       packet */
} bcm_switch_pkt_trace_info_t;
```

Returns

`BCM_E_UNAVAIL`

TRILL MANAGEMENT

A new trill port flag has been added in this release while the new trill VPN config flag `BCM_TRILL_VPN_TRANSPARENT_SERVICE` has been added for `bcm_trill_vpn_create()` API.

Table 84: TRILL port flags

Name	*Purpose*
`BCM_TRILL_MULTICAST_TRANSPARENT_SERVICE`	Create TRILL multicast transparent service

Table 85: TRILL VPN Config Flags

Name	*Purpose*
`BCM_TRILL_VPN_TRANSPARENT_SERVICE`	Create VPN with transparent service.

TRUNKING (LINK AGGREGATION)

Two Trunk PSC macros have been added in this release. The macros are designated for specifying the Port Selection Criteria (PSC), that is, how to spread the traffic across the trunk's member ports:

Table 86: Macros for Specifying Port Selection Criteria (PSC)

Macro	*PSC Description*
`BCM_TRUNK_PSC_RANDOMIZED`	Random load balancing selection of members.
`BCM_TRUNK_PSC_SMOOTH_DIVISION`	Use Smooth Division table to choose members.

VLAN MANAGEMENT

A new field `aging_cycles` has been added into data structure `bcm_vlan_control_vlan_t`, while `BCM_VLAN_PORT_VSI_BASE_VID` flag has been added as one of `bcm_vlan_port_t` flags field for the combination of one or more of the flags that can be passed to `bcm_vlan_port_create()` API. The `BCM_VLAN_PORT_MATCH_PORT_UNTAGGED` has also been added as one of port match criteria for a Logical Layer 2 port that can be set as one of the values from `bcm_vlan_port_match_t`.

Two VLAN actions have been added in the defines of `BCM_VLAN_ACTION_FLAGS` for `bcm_vlan_action_set_t`.

```
typedef struct {
    ...
    int                     aging_cycles; /* number of aging meta-cycles */
} bcm_vlan_control_vlan_t;
```

Table 87: bcm_vlan_port_t Flags

FLAG	*Meaning*
`BCM_VLAN_PORT_VSI_BASE_VID`	VSI assignment mode.VSI equal to VSI in LIF table + VID
`BCM_VLAN_PORT_CASCADED`	Creates a VLAN port object of type FEC that is pointed by another VLAN port object

Table 88: bcm_vlan_port_match_t

Value	*Meaning*
`BCM_VLAN_PORT_MATCH_PORT_UNTAGGED`	Mod/port/trunk for untagged packets.

Table 89: VLAN action flags

Flag	Value	Description
BCM_VLAN_ACTION_SET_TPID_ALLOW_CREATE	0x0010	Create a TPID profile if TPIDs can't match an existing one.
BCM_VLAN_ACTION_SET_TPID_ALLOW_DESTROY	0x0020	Destroy a TPID profile using specifi TPIDs.

UDF RESOURCES MANAGEMENT

A new UDF option for UDF creation has been added.

Table 90: Options to bcm_udf_create

flags	Description
BCM_UDF_CREATE_O_POLICER_GROUP	UDF allocation will be adjusted for use by global (service) meter.

Section 4: Test Statistics

HOW TO READ THE DATA

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

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:

Table 91:

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.

NOTE

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 API.

Finally, some devices have fewer columns listed if they were introduced recently.

Note on the suite titled "DVAPI": As of 6.3.6, the test suite labeled "DVAPI" has been split up into many sub-modules. This was done as a means to provide much clearer visibility both internally and externally for the quality of each device going forward. Because this change is not retroactive for past release data, data for previous releases will still contain the "DVAPI" suite. Level of quality can still be discerned via the other modules as well as the aggregate level (and it should be noted that even more tests have been added in 6.3.6, so there was no reduction in coverage). Future releases will continue to use the higher-granularity format.

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.

Table 92:

	sdk-6.4.2	sdk-6.4.1	sdk-6.3.9	sdk-6.2.9
golden	154	154	154	154
warmboot	294	288	290	254
auth	17	17	17	N/A
bfd	37	37	37	16
bhh	43	15	43	15
chip	9	9	9	N/A
cint	76	55	55	52
coe	510	510	510	N/A
cosq	310	306	300	289
custom	7	7	7	N/A
dvapi*	N/A	N/A	N/A	985
ea	108	108	108	108
eav	19	19	19	N/A
extender	12	12	12	N/A
fabric	7	7	7	N/A
failover	8	8	8	N/A
fcoe	37	37	37	N/A
field	760	734	732	704
higigproxy	129	129	129	N/A
infra	114	114	114	114
ipfix	17	17	17	N/A
ipmc	63	56	59	N/A
l2	238	230	238	222
l2gre	13	13	13	N/A
l3	262	241	255	211
l3.alpm.combined	67	63	67	N/A
l3.alpm.combined.64	55	51	67	N/A
l3.alpm.parallel	67	63	63	N/A
l3.alpm.parallel.64	55	51	51	N/A
link	26	26	26	N/A
mim	19	19	19	N/A
mirror	146	146	146	N/A
misc	16	16	16	N/A
mpls	128	128	127	80
multicast	17	17	17	N/A
niv	23	13	13	N/A
oam	187	133	187	N/A
pkt	44	44	44	N/A
port	262	261	262	N/A
proxy	23	23	23	N/A
ptp	115	115	115	1
qos	12	12	12	N/A

Table 92:

	sdk-6.4.2	sdk-6.4.1	sdk-6.3.9	sdk-6.2.9
rate	21	21	21	N/A
rtag7	32	24	32	6
rx	21	21	21	N/A
ser	52	52	52	N/A
stack	49	49	49	49
stat	203	203	202	65
stg	42	42	42	N/A
switch	189	130	199	N/A
time	16	16	16	N/A
tlvMsg	13	13	13	13
trill	40	40	40	36
trunk	177	173	173	139
tunnel	65	65	65	65
subport	30	33	30	12
vlan	207	207	201	186
vxlan	78	69	78	41
wlan	17	17	17	N/A
Test Suite Total	5758	5479	5676	3817

*Note on DVAPI: The DVAPI test suite has been split into many sub-module for higher visibility. Please see the "Note on the suite titled "DVAPI" in the Notes section above for more information.

TEST RESULTS

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

ALL DEVICES

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

Table 93:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*	*sdk-6.2.9*
golden	1.1 %	1.5 %	0.7 %	5.1 %
warmboot	1.2 %	1.5 %	1.6 %	8.2 %
bcm.auth	0.4 %	0.2 %	0.0 %	N/A
bcm.bfd	0.0 %	0.9 %	0.0 %	0.0 %
bcm.bhh	1.6 %	2.0 %	0.0 %	5.5 %
bcm.chip	1.5 %	1.7 %	0.7 %	N/A
bcm.cint	1.3 %	0.0 %	0.0 %	13.0 %
bcm.coe	0.1 %	0.4 %	0.1 %	N/A
bcm.cosq	0.8 %	1.9 %	1.0 %	3.2 %
bcm.custom	0.0 %	0.0 %	0.0 %	N/A
bcm.dvapi	N/A	N/A	N/A	3.4 %
bcm.ea	0.0 %	0.0 %	0.0 %	0.2 %
bcm.eav	0.0 %	0.0 %	0.0 %	N/A
bcm.extender	0.0 %	0.0 %	0.0 %	N/A
bcm.fabric	0.0 %	0.0 %	0.0 %	N/A
bcm.failover	0.0 %	0.0 %	0.0 %	N/A
bcm.fcoe	0.2 %	0.1 %	0.1 %	N/A
bcm.field	1.7 %	1.6 %	0.9 %	2.2 %
bcm.higigproxy	0.5 %	0.9 %	0.5 %	N/A
bcm.infra	0.1 %	0.0 %	0.2 %	0.0 %
bcm.ipfix	0.5 %	0.7 %	0.8 %	N/A
bcm.ipmc	0.4 %	0.8 %	0.4 %	N/A
bcm.l2	1.0 %	1.5 %	1.2 %	1.8 %
bcm.l2gre	0.3 %	0.0 %	0.0 %	N/A
bcm.l3	1.0 %	1.0 %	0.8 %	3.1 %
bcm.l3.alpm.combined	0.0 %	0.0 %	0.0 %	10.4 %
bcm.l3.alpm.combined.64	0.0 %	0.0 %	0.0 %	N/A
bcm.l3.alpm.parallel	0.0 %	0.0 %	0.0 %	10.2 %
bcm.l3.alpm.parallel.64	0.0 %	0.0 %	0.0 %	N/A
bcm.link	0.0 %	0.1 %	0.0 %	N/A
bcm.mim	0.0 %	0.1 %	0.0 %	N/A
bcm.mirror	2.2 %	2.8 %	3.0 %	N/A
bcm.misc	0.7 %	0.7 %	0.8 %	N/A
bcm.mpls	0.4 %	0.6 %	0.4 %	1.3 %
bcm.multicast	0.4 %	2.6 %	0.3 %	N/A
bcm.niv	0.3 %	0.1 %	0.0 %	N/A
bcm.oam	0.0 %	1.1 %	0.0 %	N/A
bcm.pkt	0.0 %	0.0 %	0.0 %	N/A

Table 93:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*	*sdk-6.2.9*
bcm.port	1.4 %	1.5 %	1.2 %	N/A
bcm.proxy	1.1 %	0.7 %	0.0 %	N/A
bcm.ptp	0.0 %	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %	N/A
bcm.rate	0.0 %	0.9 %	0.0 %	N/A
bcm.rtag7	0.1 %	0.0 %	0.0 %	0.2 %
bcm.rx	0.1 %	0.9 %	0.1 %	N/A
bcm.ser	0.3 %	0.8 %	0.2 %	N/A
bcm.stack	0.2 %	0.2 %	0.1 %	1.0 %
bcm.stat	0.8 %	1.0 %	0.6 %	4.5 %
bcm.stg	0.5 %	0.3 %	0.6 %	N/A
bcm.switch	0.6 %	1.0 %	0.6 %	N/A
bcm.time	0.0 %	0.0 %	0.0 %	N/A
bcm.tlvMsg	0.0 %	0.0 %	0.0 %	0.0 %
bcm.trill	0.9 %	1.1 %	0.5 %	10.5 %
bcm.trunk	0.9 %	2.0 %	1.1 %	1.4 %
bcm.tunnel	0.3 %	0.0 %	0.0 %	0.2 %
bcm.subport	0.8 %	1.7 %	1.3 %	36.5 %
bcm.vlan	0.6 %	1.2 %	0.9 %	1.4 %
bcm.vxlan	0.0 %	0.1 %	0.0 %	2.2 %
bcm.wlan	0.8 %	1.8 %	1.0 %	N/A
Test Suite Total	0.8 %	1.1 %	0.7 %	2.8 %

*Note on DVAPI: The DVAPI test suite has been split into many sub-module for higher visibility. Please see the "Note on the suite titled "DVAPI" in the Notes section above for more information.

TRIDENT2

Table 94:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*	*sdk-6.2.9*
golden	0.1 %	0.1 %	0.0 %	6.5 %
warmboot	1.5 %	1.2 %	1.0 %	11.3 %
bcm.auth	0.0 %	0.0 %	0.0 %	N/A
bcm.bfd	0.0 %	0.0 %	0.0 %	0.0 %
bcm.bhh	N/A	N/A	N/A	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %	N/A
bcm.cint	1.3 %	0.0 %	0.0 %	13.5 %
bcm.coe	0.0 %	0.0 %	0.0 %	N/A
bcm.cosq	0.3 %	0.7 %	0.3 %	3.1 %
bcm.custom	0.0 %	0.0 %	0.0 %	N/A
bcm.dvapi	N/A	N/A	N/A	2.7 %
bcm.ea	0.0 %	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %	N/A
bcm.extender	0.0 %	0.0 %	0.0 %	N/A
bcm.fabric	0.0 %	0.0 %	0.0 %	N/A
bcm.failover	0.0 %	0.0 %	0.0 %	N/A
bcm.fcoe	2.2 %	2.7 %	2.7 %	N/A

Table 94:

	sdk-6.4.2	sdk-6.4.1	sdk-6.3.9	sdk-6.2.9
bcm.field	0.6 %	0.9 %	0.7 %	1.1 %
bcm.higigproxy	0.6 %	0.6 %	0.8 %	N/A
bcm.infra	0.5 %	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %	N/A
bcm.ipmc	0.5 %	3.1 %	0.0 %	N/A
bcm.l2	0.1 %	0.8 %	0.0 %	0.0 %
bcm.l2gre	0.9 %	0.0 %	0.0 %	N/A
bcm.l3	1.6 %	0.6 %	2.4 %	0.5 %
bcm.l3.alpm.combined	0.0 %	0.0 %	0.0 %	N/A
bcm.l3.alpm.combined.64	0.0 %	0.0 %	N/A	N/A
bcm.l3.alpm.parallel	0.0 %	0.0 %	0.0 %	N/A
bcm.l3.alpm.parallel.64	0.0 %	0.0 %	0.0 %	N/A
bcm.link	0.0 %	0.0 %	0.0 %	N/A
bcm.mim	0.0 %	0.0 %	0.0 %	N/A
bcm.mirror	0.0 %	0.0 %	0.0 %	N/A
bcm.misc	0.0 %	0.0 %	1.8 %	N/A
bcm.mpls	0.0 %	0.8 %	0.0 %	1.3 %
bcm.multicast	1.1 %	0.0 %	0.0 %	N/A
bcm.niv	1.4 %	0.0 %	0.0 %	N/A
bcm.oam	0.0 %	0.8 %	0.0 %	N/A
bcm.pkt	0.0 %	0.0 %	0.0 %	N/A
bcm.port	0.8 %	1.3 %	0.8 %	N/A
bcm.proxy	1.1 %	0.0 %	0.0 %	N/A
bcm.ptp	0.0 %	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %	N/A
bcm.rate	0.0 %	0.0 %	0.0 %	N/A
bcm.rtag7	0.0 %	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %	N/A
bcm.ser	2.4 %	3.8 %	1.9 %	N/A
bcm.stack	0.5 %	0.3 %	0.0 %	0.0 %
bcm.stat	1.0 %	0.5 %	0.0 %	3.1 %
bcm.stg	0.0 %	0.0 %	0.0 %	N/A
bcm.switch	0.7 %	1.1 %	1.0 %	N/A
bcm.time	0.0 %	0.0 %	0.0 %	N/A
bcm.tlvMsg	0.0 %	0.0 %	0.0 %	0.0 %
bcm.trill	2.0 %	0.0 %	0.0 %	25.0 %
bcm.trunk	0.9 %	2.0 %	0.6 %	1.4 %
bcm.tunnel	0.0 %	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %	50.0 %
bcm.vlan	0.3 %	0.0 %	0.0 %	1.1 %
bcm.vxlan	0.4 %	0.0 %	0.0 %	24.4 %
bcm.wlan	0.0 %	0.0 %	0.0 %	N/A
Test Suite Total	0.5 %	0.7 %	0.4 %	2.5 %

*Note on DVAPI: The DVAPI test suite has been split into many sub-module for higher visibility. Please see the "Note on the suite titled "DVAPI" in the Notes section above for more information.

TRIUMPH3

Table 95:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*	*sdk-6.2.9*
golden	1.2 %	0.2 %	0.4 %	8.9 %
warmboot	1.4 %	1.4 %	1.4 %	8.4 %
bcm.auth	0.0 %	0.0 %	0.0 %	N/A
bcm.bfd	0.0 %	0.0 %	0.0 %	0.0 %
bcm.bhh	1.4 %	2.2 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %	N/A
bcm.cint	1.3 %	0.0 %	0.0 %	13.5 %
bcm.coe	0.0 %	0.0 %	0.0 %	N/A
bcm.cosq	0.3 %	1.0 %	0.3 %	3.3 %
bcm.custom	0.0 %	0.0 %	0.0 %	N/A
bcm.dvapi	N/A	N/A	N/A	3.8 %
bcm.ea	0.0 %	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %	N/A
bcm.extender	0.0 %	0.0 %	0.0 %	N/A
bcm.fabric	0.0 %	0.0 %	0.0 %	N/A
bcm.failover	0.0 %	0.0 %	0.0 %	N/A
bcm.fcoe	0.0 %	0.0 %	0.0 %	N/A
bcm.field	1.4 %	3.6 %	1.5 %	4.6 %
bcm.higigproxy	0.8 %	0.8 %	0.7 %	N/A
bcm.infra	0.0 %	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %	N/A
bcm.ipmc	0.0 %	0.0 %	0.0 %	N/A
bcm.l2	1.8 %	1.3 %	0.8 %	2.3 %
bcm.l2gre	0.0 %	0.0 %	0.0 %	N/A
bcm.l3	0.6 %	0.4 %	0.7 %	4.8 %
bcm.link	0.0 %	0.0 %	0.0 %	N/A
bcm.mim	0.0 %	0.0 %	0.0 %	N/A
bcm.mirror	0.0 %	0.0 %	0.0 %	N/A
bcm.misc	1.9 %	2.3 %	6.3 %	N/A
bcm.mpls	0.7 %	1.4 %	0.0 %	0.3 %
bcm.multicast	0.0 %	5.9 %	0.0 %	N/A
bcm.niv	0.0 %	0.0 %	0.0 %	N/A
bcm.oam	0.0 %	0.8 %	0.0 %	N/A
bcm.pkt	0.0 %	0.0 %	0.0 %	N/A
bcm.port	2.0 %	2.4 %	3.8 %	N/A
bcm.proxy	0.0 %	0.5 %	0.0 %	N/A
bcm.ptp	0.0 %	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %	N/A
bcm.rate	0.0 %	3.3 %	0.0 %	N/A
bcm.rtag7	0.0 %	0.0 %	0.0 %	0.0 %
bcm.rx	0.9 %	0.0 %	1.1 %	N/A
bcm.ser	0.0 %	0.0 %	0.0 %	N/A
bcm.stack	0.8 %	0.7 %	2.5 %	4.8 %
bcm.stat	1.1 %	0.5 %	0.5 %	17.6 %
bcm.stg	0.0 %	0.0 %	0.0 %	N/A
bcm.switch	0.4 %	1.1 %	0.8 %	N/A

Table 95:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*	*sdk-6.2.9*
bcm.time	0.0 %	0.0 %	0.0 %	N/A
bcm.tlvMsg	0.0 %	0.0 %	0.0 %	0.0 %
bcm.trill	3.2 %	3.1 %	6.3 %	27.8 %
bcm.trunk	0.6 %	0.8 %	0.6 %	1.9 %
bcm.tunnel	0.0 %	0.0 %	0.0 %	0.8 %
bcm.subport	0.0 %	0.0 %	0.0 %	42.4 %
bcm.vlan	0.0 %	0.0 %	0.2 %	2.6 %
bcm.vxlan	0.0 %	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	3.2 %	0.0 %	N/A
Test Suite Total	0.6 %	1.5 %	0.6 %	4.1 %

*Note on DVAPI: The DVAPI test suite has been split into many sub-module for higher visibility. Please see the "Note on the suite titled "DVAPI" in the Notes section above for more information.

HURRICANE2

Table 96:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*
golden	0.1 %	0.2 %	0.0 %
warmboot	0.0 %	2.1 %	1.7 %
bcm.auth	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	1.3 %	0.0 %	0.0 %
bcm.coe	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.0 %	0.3 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.4 %	0.7 %	0.4 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.5 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.3 %	0.3 %	0.2 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.4 %	0.4 %	0.4 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %

Table 96:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	1.2 %	1.7 %	1.2 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.1 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.5 %	1.5 %	0.5 %
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	0.1 %	0.5 %	0.6 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	1.6 %	1.5 %	1.5 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.3 %	0.4 %	0.3 %

*Note on DVAPI: The DVAPI test suite has been split into many sub-module for higher visibility. Please see the "Note on the suite titled "DVAPI" in the Notes section above for more information.

HELIX4

Table 97:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*
golden	0.9 %	0.3 %	0.6 %
warmboot	0.3 %	2.4 %	0.5 %
bcm.auth	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	1.3 %	0.0 %	0.0 %
bcm.coe	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.7 %	0.2 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.dvapi	N/A	N/A	N/A
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	0.3 %	1.5 %	0.6 %

Table 97:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*
bcm.higigproxy	0.8 %	0.8 %	0.8 %
bcm.infra	0.7 %	0.0 %	1.5 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.4 %	0.0 %
bcm.l2	0.6 %	0.4 %	0.4 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.1 %	0.4 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	1.4 %	1.3 %	1.5 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	4.8 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.5 %	0.5 %	0.5 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.6 %	0.6 %	0.6 %
bcm.tunnel	0.0 %	0.0 %	0.6 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.6 %	0.5 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.2 %	0.6 %	0.4 %

*Note on DVAPI: The DVAPI test suite has been split into many sub-module for higher visibility. Please see the "Note on the suite titled "DVAPI" in the Notes section above for more information.

KATANA2

Table 98:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*
golden	0.2 %	1.1 %	0.0 %
warmboot	0.6 %	1.4 %	1.4 %

Table 98:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*
bcm.auth	0.0 %	0.0 %	0.0 %
bcm.bfd	0.0 %	0.0 %	0.0 %
bcm.bhh	1.4 %	5.4 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	1.3 %	0.0 %	0.0 %
bcm.coe	0.7 %	1.7 %	1.6 %
bcm.cosq	0.3 %	1.9 %	0.5 %
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	0.3 %	0.9 %	0.7 %
bcm.higigproxy	0.2 %	0.4 %	0.4 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.2 %	0.0 %	0.0 %
bcm.l2	0.4 %	1.0 %	0.4 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	1.1 %	1.5 %	1.4 %
bcm.link	0.0 %	1.1 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.8 %	0.0 %	1.6 %
bcm.mpls	1.8 %	2.6 %	1.6 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	2.4 %	2.8 %	2.2 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	1.9 %	2.9 %	1.4 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.5 %	0.5 %	0.5 %
bcm.stg	0.0 %	0.3 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.4 %	0.4 %	0.4 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	1.0 %	12.1 %	1.7 %
bcm.vlan	0.3 %	0.0 %	0.2 %
bcm.vxlan	0.0 %	0.0 %	0.0 %

Table 98:

	sdk-6.4.2	*sdk-6.4.1*	*sdk-6.3.9*
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.6 %	1.6 %	0.6 %

*Note on DVAPI: The DVAPI test suite has been split into many sub-module for higher visibility. Please see the "Note on the suite titled "DVAPI" in the Notes section above for more information.

STATIC CODE QUALITY ANALYSIS

Continued progress in whittling down static analysis defects per plan.

Table 99:

	Initial Reported Issues	***Reported Issues SDK 6.4.1***	***Reported Issues SDK 6.4.2***
DNX	664	131	67
XGS	271	123	54
SBX	600	0	0
SerDes	147	74	21
Common	2827	200	60
Total	4509	528	202

Section 5: Resolved Issues for 6.4.2

The following issues are resolved in version 6.4.2 of the SDK.

Table 100:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.2*
SDK-31854		All	The set and get versions of `bcm_rx_redirect_reasons` have been updated to return the correct error codes for wrong unit or wrong mode.
SDK-45222	614537	56440_B0	`mis_con_mep_id` and length fields are added in endpoint info structure
SDK-45223	614521	56440_B0	Problem: If user configured, source `mep_id` on both DUTs differently, than the BFD session was not able to be established properly intermittently. To debug this the user user requested that the SW display/validate the Remote DUT `mep_id`, which was configured incorrectly. Solution: New fields are added in BFD endpoint info as `remote_mep_id` and `mis_con_mep_id` to achieve this. After enabling this feature, user needs to configure on both DUTs, `mep_id` and `remote_mep_id` and and as well as length fields. If user mep configuration is incorrect, wrong mep id will be stored in `mis_con_mep_id.` From BCM shell, user has to invoke `bcm_bfd_endpoint_get()` API to display wrong mep id.
SDK-45253		56643_B0	For flexible HG ports port mode register xport field should be set to 0 (Single mode). Setting it to zero removes the underflow issue. Fixed the same.
SDK-46712	634930	56840_A0	MC Prio2Cos values have been corrected to reset properly
SDK-47231	643758	56440_A0 56445_A0 56440_A1 56445_A1 56444_A1 56449_B0 56445_B0 56440_B0 56447_B0 56443_B0 56441_B0 56446_B0 56448_B0	In the previous release packet was not getting received while switching from IEEE mode to HIGIG mode. This was because the `STRICT_PREAMBLE` and `PROCESS_VARIABLE_PREAMBLE` in `XMAC_RX_CTRL` was not configured correctly for HG port. This issue has now been addressed by setting `STRICT_PREAMBLE=0` and `PROCESS_VARIABLE_PREAMBLE=0` in `XMAC_RX_CTRL` when port type is HG.
SDK-48892	652937	56850_A0	Fixed HSP port attach for Y pipe based ports.
SDK-49048		88650_A0	Qpairs allocation of CPU ports is done from a specific qpairs range of 192-255. With this fix the qpairs allocation is done according to CPU channel. For each CPU port the following range of qpairs is allocated: [192+channel, `192+channel+nof_priorities-1]`.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-49169		56850_A0 56850_A1	fix code to display correctly ecmp info for host route.
SDK-49620	684645	56850_A0	Fixed addition of a member to an empty LAG, Higig trunk or ECMP group, when resilient hashing is enabled. This fix is applicable to BCM56850 device.
SDK-49630		56850_A0 56850_A1 56850_A2	Fixed to retain the valid actions when unwanted action is removed
SDK-49910		56640_A0 56640_A1 56640_B0	The issue is that the function `_bcm_tr3_cosq_node_get` expects a GPORT node to be passed. However, for the cpu node we don't pass a gport and this returns `BCM_E_PORT.` The fix is to get the sched index via `_bcm_tr3_cosq_index_resolve` in the alternate path.
SDK-50311	693316	56850_A0 56850_A1 56850_A2	In earlier releases, the mask in flex hash entry could not be configured with existing API. This has been resolved.
SDK-50689		56640_A0 56640_A1 56640_B0	Index return for TCAM search cli was not correct while searching data in L2 tcam partition. Rectified this behavior for tcam search cli and corrected index returned for searched data in L2 tcam partition.
SDK-51769	691762	56840_A0	In earlier release, when HIGIG port speed was 40G and external phy was in reverse mode, the SOP, SOM and sequence alignment were not programmed correctly. This has been resolved.
SDK-52237	721240	All	Support to handle concurrent RPC requests is added . New Configuration variable `"rpc_server_thread_count"` is added. This variable holds the number of RPC server threads created.
SDK-52679	726763	56450_A0	Problem Statement: Previously we could not support movement for L0 node from one port to other port Solution: SDK provided a new API, `bcm_cosq_gport_reattach` which will allow user to move L0 node from one port to other
SDK-52710	729008	56224_B0	On UNIMAC devices, calling `bcm_stat_init()` API was overwriting the maximum frame setting on port when configuration for Jumbo packet handling using `bcm_stat_jumbo.` This configuration will set the size used by counter collection to decide when to count over-sized packets. When call `bcm_stat_init()` the maximum frame size was being set to this jumbo size setting ovewriting the existing max frame setting on a port. Removed this code for setting the max frame size during `bcm_stat_init.` THe counter collection for over-sized packet has been fixed to take into account the oversize stats.
SDK-53222 SDK-52915	733469	56450_A0	In Katana2 the capability to display the lls tree of the coe port was missing and It has been added

Table 100:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.2*
SDK-53223	733490	`56450_A0`	Shaper configuration was not reset when the nodes in scheduler tree were deleted or detached in the tree. Added code to reset the shaper configuration when the nodes in scheduler tree are deleted or detached.

SDK-54341 SDK-57678		`56640_A0` `56344_A0`	When changing the parent of any L2 node from WDRR to SP, `BCM_E_TIMEOUT` is thrown, since dynamic change of SP<->WRR is not supported. The current way to resolve this is: Customer calls APIs to stop enqueue and drain the mmu queues before changing the scheduler mode.
			With this method, the customer is expected to call the following APIs to stop the enqueue and drain the mmu before detaching/attaching to a parent with different scheduler mode. `bcm_port_control_set(unit,` port, bcmPortControlMmuTrafficEnable, 0);
			`bcm_port_control_set(unit,` port, bcmPortControlMmuDrain, 1);
			`<Detach/attach/sched_set>`
			`bcm_port_control_set(unit,` port, bcmPortControlMmuTrafficEnable, 1);

SDK-54758	750349	`56450_A0`	Problem statement: `bcm_cos_mapping_set` is not working for flex ports Fix description: Added code so that flex ports are also used during `config_set` and `mapping_set` profiles, as a result `bcm_cos_mapping_set` is working appropriately
SDK-54796	751218	`88650_A0` `88650_B0` `88650_B1`	When allocating a combination of 1-priority and more than 1-priority ports that cover the entire q-pairs range (256 q-pairs), 1 -priority port can't be allocated on the last q-pair. The driver allocates the qpairs in ports order, so if last port is 1-priority port error will be returned. To avoid this issue user can allocate one of the ports which aren't 1-priority to use the last qpair(s). The soc property is: `otm_base_q_pair_<port_id>=<base_q_pair>`
SDK-54889	750468	`88030_A0`	Default route disappeared is caused by the bucket merge was not supported by taps driver. After we improve the taps error handling and support to merge the bucket in that handling, this issue could be resolved.
SDK-54991	750024	All	Added support for filter callbacks in the Linux KNET kernel driver.
SDK-55162		`88660_A0` `88670_A0`	IP Routing-Over-Overlay (ROO) refers to a set of protocols/applications where the L2 forwarding to the Host/Next-Hop router is not accomplished by simple 802.1q bridging, but by L2-Overlay protocols (VXLAN, etc). BCM8866X supports ROO Host Unicast over VXLAN. See `cint_vxlan_roo.c` for cint example and Programmer's Reference Guide for more details.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-55180		All	In non-interface map mode, `bcm_l3_intf_create()` and `bcm_vlan_control_vlan_set()` can overwrite vrf values set by each other. This is how these apis are designed to set up the parameters that they are overstepping each other. Added documentation to explain the behavior of APIs in case of VRF setting.
SDK-55184		`56850_A0`	Earlier SDK releases did not allow configuring MTU value for vxlan access ports. This release now supports setting/resetting MTU for vxlan access ports through `bcm_vxlan_port_add()` API.
SDK-55454	758146	`88650_A0`	In Counter Processor, one of the Ingress-PP counting mode is VSI. A support for counting packets by ingress VSI is added. The relevant soc property is `counter_engine_source_<counter ID>=INGRESS_VSI.` When set, the default Counter-ID value for Ethernet packets is VSI. This value may be overridden by Ingress PMF (Field Processor APIs with bcmFiedlActionStat0/1 actions) or by Rx-traps.

SDK-55607	752009	`56640_A0` `56540_A0`
		`56640_A1` `56640_B0`
		`56540_B0`
			When OAM LM endpoint is deleted on Triumph3, feild group is also deleted if there are no more OAM LM endpoints present in that group. On deleting the field group, field group index was not set to invalid. This issue has been fixed.

SDK-55706	730810	`56540_A0` `56540_B0`	Issue :- "fp show" and "listmem `fp_tcam"` are reporting incorrect entry values for IFP in Apollo2 Fix :- Updated "fp show" command to return correct entry count for IFP in Apollo2. "ListMem `fp_tcam"` cannot be fixed as the command reads from driver. Modifying the driver values is leading to other issues.
SDK-55886	760333	All	In previous releases, out-of-bounds access to an array of multicast group was allowed, which would result in crashing. This has been resolved by placing a proper check to ipmc id.
SDK-55929	759404	`56620_B0`	In earlier SDK releases, the SDK did not properly decode the dest modid and dest port in the Triumph and Triumph2 class of devices. The issue is fixed by appropriately accounting for the addressable port and module id range on the device.
SDK-56001	761280	All	`soc_feature_timesync_support` flag enables 1588/PTP protocol's transparent clock support. This JIRA enables PTP transparent clock support for Triumph 3.

Table 100:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.2*
SDK-56024		`56850_A0`	There's a bit in the `VLAN_XLATE` table called `VLAN_ACTION_VALID, It` must be enabled to process `XLATE__DISABLE_VLAN_CHECKS` for VXLAN virtual ports, but disable it for VXLAN access ports to drop packets at ingress. They have conflict. To solve the problem, a new flag has been added that allows the customer to control the bit, The new flag is `BCM_VXLAN_PORT_ENABLE_VLAN_CHECKS.`
SDK-56065 SDK-61361		`88670_A0`	In BCM88660 and below, any FEC has both a destination and a forwarding information (e.g., Out-LIF). The destination cannot be another FEC. In BCM88670, the destination of a FEC can be another FEC (Hierarchical FEC). There is an hardware limitations on this feature - a pointing FEC may not be accessed in the same bank as the pointed FEC (In hardware the FECs have been divided into banks). Due to this limitation the FECs must be allocated in such a way that a pointing FEC and a pointed FEC will never be in the same bank. A CASCADED flag has been created for any API that allocates FECs `(bcm_l3_egress_create, bcm_vlan_port_create, bcm_mpls_port_create)` to indicate that a FEC will be pointed by another FEC. If the CASCADED flag is not specified, the FEC can be used as a regular FEC or a pointing FEC. For more details see the Packet Processing User Manual.
SDK-56099	766312	`56334_B0 56334_A0`	Repaired `EGR_VLAN` table corruption during hash table entry balancing in Enduro devices.
SDK-56203		`88650_A0 88650_B0 88650_B1 88660_A0`	XLPORT Overrun/Underrun Workaround ---: The Arad driver implements a sequence to recognize and recover the port from XLPORT Overrun/ Underrun issue (see BCM88650 errata sheet). To activate the sequence during device init use the following soc property: `custom_feature_nif_recovery_enable=1` (default is disabled on 6.3.x, and enabled on 6.4.x). The sequence might perform several iterations when trying to recover the port. To limit number of iteration use the following SoC property: `custom_feature_nif_recovery_iter` (default is 3). Note that from lab experience the port is recover within single iteration. Limitations: 1. The SW WA works for XLP0 only. 2. The SW WA is called during init and isnt available for dynamic port.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-56317		56846_A0 56846_A1	In previous releases, created multipaths more than max capacity could corrupt existing ECMP groups and return wrong value -1 if ECMP group size of TD+ configured to 256 as TD device. In this release, it returns BCM_E_FULL(-6) if creating ECMP multipaths more than max capacity.
SDK-56424	767587	56850_A0	Problem: Modified srtcm meter mode support was missing in Trident2 for Egress FieldProcessor Stage Solution: Added modified srtcm metering support in coldboot and warmboot for Egress Field Processor Stage in Trident2.This includes policer creation, attach and install the configs. It also includes recovery of the meter during warmboot.
SDK-56462	770876	56640_A0 56640_A1 56640_B0	Added the software support for 100G Remote loopback
SDK-56589	771496	56450_B0 56455_A0 56456_A0 56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56640_A0 56440_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56843_B0 56841_A3 56445_A0 56440_A1 56445_A1 56444_A1 56450_A0 56846_A1 56841_B0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0 56449_B0 56445_B0 56440_B0 56447_B0 56443_B0 56441_B0 56446_B0 56448_B0 56442_B0	Customer found the counter RDVLN did not work without configuring XMAC_RX_VLAN_TAG register. They also found there was no way to set this register in previous SDK release. Now an improvement has been implemented to set the register by bcm_port_tpid_set() & bcm_port_inner_tpid_set() APIs.
SDK-56619	771355	56846_A1	Issue: In earlier releases, no fault message was sent in single fiber connection for issue in phy_wc40_software_rx_los(). Root Cause: The rx_los_state was never set to RESET. So LOCAL_FAULT_ENABLE was never set as the RESET case was never hit. Fix: Allow the rx_los_state to set to RESET and subsequently move into the RESET case to set rx_los_state to START_TIMER.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-56776	772447	56440_A0 56440_B0 56450_A0 56450_B0 56850_A0	Issue: When DK_MEM flag is enabled, we would access L2X views array during L2 initialization. We would use KEY_TYPE as index in L2X views array. KEY_TYPE field is 3 bits wide in KT/KT2 and 4 bits wide in TD2. Hence it is set to 7 in KT/KT2 and 15 in TD2. For KT,KT2 and TD2, KEY_TYPE is greater than maximum index of L2X views array and it may result in segmentation fault while accessing this array with index greater than maximum index. Fix: In soc_mem_entry_dump_common, for KT/KT2/TD2, if key_type is greater than maximum index of L2X views array, then key_type value is set to the respective L2X view array maximum index value.
SDK-56787	775384	All	In earlier releases, there is no way to use the API bcm_rate_bandwidth_set to set the rate to 0kbps speed. If the pass the speed as zero, the assumption was to disable metering. Now, to disable metering, kbits_sec parameter has to be passed as BCM_RATE_DISABLE and to set the rate to zero, kbits_sec parameter has to be passed as BCM_RATE_BLOCK
SDK-56916	775184	56340_A0	Problem: Regex match was not being reported if the pattern occurs beyond 2048 bytes for any given session. The reason was that the AXP_SM_SIGNATURE_MATCH_CONTROL.MAX_BYTE_INSPECTED is always being set to the default value of 2048. Solution: Set the above field/register to the payload depth provided through bcm_regex_config_set API. This will be set in sync with the FT_CONFIG.MAX_BYTES field.
SDK-57000	755118	56850_A0 56850_A1 56850_A2	Support has been added for VPLAG for VxLAN based VPs.
SDK-57024		88650_A0	Dynamic port provisioning feature added
SDK-57146	779181	56850_A0 56850_A1 56850_A2	Description: MAP_TABLE is not correctly indexed in bcm_esw_switch_rcpu_encap_priority_map_set() Root cause: It's a typo. Solution: The software now corrects the typo.
SDK-57171	780263	56850_A0 56850_A1 56850_A2	Originally, VLAN_XLATE entry was not cleared possibly when deleting a VP. It is fixed by setting modid or trunk_id to an invalid value (-1) in _bcm_td2_vxlan_match_add when creating a vxlan logical port on a physical port or a trunk.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-57197	776292	56620_B0	Description: A parity entry in L3_DEFIP_DATA_ONLY can't be cleared by SDK-6.3.6 Root cause: In the earlier software, the table L3_DEFIP_DATA_ONLY is not cached, so the SER engine returns not available, and hence the corrupted entry would not be cleared. Solution: Current software has the SOC_MEM_FLAG_SER_CACHE_RESTORE set to L3_DEFIP_DATA_ONLY, once an SER error was detected on an entry of L3_DEFIP_DATA_ONLY, the ser correction engine will clear/restore the whole L3_DEFIP entry.
SDK-57205	779120	56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56843_B0 56841_A3 56846_A1 56841_B0	In previous release, bcm_stat_get/clear() APIs could only handle software counters on CPU port although chip has hardware CPU counters. Now these APIs have been improved to support hardware CPU counters. This improvement only works on TD, TD2, TH chips now.
SDK-57207	777630	56640_A0 56640_A1 56640_B0	Issue:Packets of size 64 to 75 bytes getting dropped for XE ports. Root Cause: The runt threshold value for XE ports was getting set as 76 instead of the correct value 64. Hence packets of size 64-75 bytes were getting dropped. Fix: For Triumph3 and Katana2, put explicit checks to ensure that runt threshold value is set to correct value, i.e. RUNT_THRESHOLD_XE = 64, RUNT_THRESHOLD_GE = 64 and RUNT_THRESHOLD_HG = 76. Also optimized the function mac_x_init for multiple READ and WRITE for XMAC_RX_CTRL and XMAC_TX_CTRL. Added a single write common for all devices instead of multiple instances as was present previously.
SDK-57250		88660_A0	PON ONU tunnel-ID allocation: So far PON application reserved last 512 tunnel-id entries of PON port 7 by ARAD. An error occurs when trying to set tunnel ID profile mapping. Fixed mapping to allow up to 2016 tunnels-IDs (0-2015) for all PON ports. TX behavior has been changed since this improvement. PTCHs(packet termination control header) of packets sent from CPU should always have SSP(source system port).
SDK-57306	781729	88650_A0 88670_A0	Change dram_crc_del_buffer_max_reclamis description in Arad/Jericho UM TM. No code changes.

Table 100:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.2*
SDK-57324		56640_A0	bcm_policer_group_create API creates a group of policers which are mapped based on the input offset derived from the packet. This mapping is fixed and is not flexible. New API bcm_policer_group_create_with_map provides the flexibility to offset derived from the packet and policer flexibly. This mapping can be given as input through the structure struct bcm_policer_map_t { int count; /* Number of entries in the map */ uint8 *offset_map; /* Array of policer offsets corresponding to the offset derived from the packet */ } Syntax of the new API : int bcm_policer_group_create_with_map(int unit, bcm_policer_group_mode_t mode, bcm_policer_t *policer_id, int *num_policers, bcm_policer_map_t *offset_map);
SDK-57344	780748	All 56548_A0 56546_A0 56545_A0 56544_A0 56542_A0 56541_A0 56540_A0 56545_A1 56540_B0 56541_B0 56546_B0 56544_B0 56547_A0 56545_B0 56542_B0	Fixed KNET DMA locking issue which could cause duplicate packets to be sent from the bcm_tx() API.
SDK-57527		56640_A0 56850_A0	In case of TD2, CMIC free running clock was pointing to external clock even if no external clock is attached to unit. Fix is provided to addess the issue so that internal reference clock is used if no external clock is present on reference pad.
SDK-57723	787163	56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56850_A0 56855_A0 56843_B0 56841_A3 56846_A1 56841_B0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In the previous release, the validity of niv_src_vif was not checked when calling _bcm_trident_mirror_niv_tunnel_set. In this release, the niv_src_vif is masked with _BCM_TD_MIRROR_NIV_SRC_VIF_MASK when calling _bcm_trident_mirror_niv_tunnel_set
SDK-57777		56440_B0 56450_B0	new config variable mmu_num_subscriber_queues and a macro "BCM_GPORT_UCAST_SUBSCRIBER_QUEUE_GROUP_SYSQID_SET" are defined to form the desired hierarchy tree of subscriber queues on front panel ports.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-57823	789287	56450_A0 56450_B0	Fixed the issue of access to non-existent fields in ECC error registers resulting in assertion during ECC interrupt handling by making sure that only the valid fields in the ECC error register are accessed.
SDK-57834	778524	56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0	Added support for 64bit statistics counters for BFD endpoints.
SDK-57846	788171	56840_A0 56640_A0 56850_A0 56640_A1 56640_B0 56850_A1 56850_A2	In earlier release DMA buffer used to download WC firmware was not freed upon detach or exit clean. This has been resolved.
SDK-57853		88660_A0	Trill warmboot. Sw state trill alloc link list size was not correctly calculated at warmboot trill restore, causing incorrect size after warmboot
SDK-57859	789891	56450_A0 56450_B0	Problem : Old TDM-B was not having flex capability and was not matching to customer requirement i.e. 16x2.5G + 1G(WC0) + 10G(WC1) Fix : New TDM-B is provided to meet 16x2.5G +1G(WC0) + 10G(WC1).
SDK-57905 SDK-55025	790243	56548_A0 56547_A0	Firescout had a few static modes that could be used to limit 128-v6 and v4 routes from the same shared space. This way of configuration was not flexible as the modes were limited. This improvement removes all the static modes and provides a mechanism which can help users to configure v6/v6 routes flexibly. User can setup limit at route level granularity with the new approach. When the limit is set for 128-V6 routes, then rest of the memory is given to v4/64-v6 routes. user can also configure whether to share 128-v6 routes space with v4 routes.
SDK-57983	789226	56850_A2	DMA source mode was not set after warmboot. This is fixed
SDK-58048		53125_A0 53018_A0 53125_B0	Fixed segmentation fault encountered when SDK was operated with KNET on the Northstar and Starfighter devices of Robo family.
SDK-58055	788661	56450_A0 56450_B0	Issue: Access to iproc unimac(ethernet) registers is failing. Rootcause: Access failed in iproc linux since `request_mem_region` in linux kernel returns `NULL`. `request_mem_region` would return `NULL` when the memory range for specific driver is occupied already. This is because the ethernet driver in iProc was already using the memory specified for unimac registers. Fix: After commenting out the `request_mem_region` in linux kernel, we can now read/write those registers.
SDK-58073	790115	56334_B0 56334_A0	Port's enable status would no longer influence `soc_phyctrl_loopback_set` setting.

Table 100:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.2***
SDK-58083		88650_A0 88660_A0	VLAN edit: bcm_vlan_translate_action_create() defines a VLAN edit action per LIF in a standard VLAN edit mode. The API can be provided with a physical port and inner/outer VLAN values. The API didn't fail call with VLAN out of range values (>4095). A validation was added.
SDK-58118	791236	56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56850_A0 56855_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In previous release, there was no support to configure RTAG7_MPLS_L2_PAYLOAD_L3_HASH_FIELD_BMAP.MPLS_L2_PAYLOAD_L3_BITMAP_A/B, RTAG7_HASH_CONTROL.MPLS_PAYLOAD_HASH_SELECT_A/B on Triumph3, RTAG7_HASH_CONTROL_2.MPLS_PAYLOAD_HASH_SELECT_A/B on Trident2. This has been resolved by providing and implementing bcmSwitchHashL3L2MPLSField0/1, bcmSwitchHashL2MPLSPayloadSelect0/1.
SDK-58163		56850_A2	In previous release, invoking bcm_trunk_set to a existing Trunk or HiGig Trunk would cause flow set table to be totally re-computed. This has been resolved in a manner that the flows assigned to staying members will now be kept without shuffling.
SDK-58247	787119	53018_A0	Fixed incorrect use of tx channel index of IPROC platform in KNET driver on BCM5301x platforms.
SDK-58270	790312	56150_A0	The entry status of dual hash search mechanism on Hurricane2 and Greyhound devices were incorrectly reported in the case of hash conflict.This has been fixed in this release.
SDK-58329	791806	56850_A0 56850_A1 56850_A2	In the previous release, there are some static analysis issues found by customer in the ALPM module. Now those issues have been fixed.
SDK-58485	796744	All 56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56850_A0 56855_A0 56843_B0 56841_A3 56846_A1 56841_B0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In the previous release, the internal priority could not be mapped correctly from the field DSCP in the IP packets with the API bcm_qos_port_map_set on TRIDETNT2. In this release, this issue has been addressed by programming the field TRUST_DSCP_PTR in the table PORT_TAB.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-58501	794058	56850_A0 56850_A1 56850_A2	In the previous release, bcm_fcoe_route_find didn`t support the return of a l3_intf id. In this release, it supports the return of the l3_intf id.
SDK-58525	794780	88650_A0 88650_B0 88650_B1 88660_A0 88670_A0	The corrective action for interrupt FCT_UNRCH_DEST_EVENT doesn't work for unit#0 - Fixed.
SDK-58550	797110	88750_B0	FE13 Isolation is intended for a planned removal of an FE card without affecting traffic. Executing the current isolation sequence on an FE13 while under traffic will cause traffic loss. The reason behind it is that the FE13 isolation sequence will cause the FE3 RTP table to be updated before the FE1 RTP table. Therefore traffic will still go into the fabric plane, but the FE3 will not be able to forward it, and will be forced to drop the traffic. Workaround: Isolate an entire FE plane, by masking the relevant links in the FDT across all the FAPs in the system. Then execute the isolation sequence on the FE13 card. This option will stop all data cells from running through the fabric, allowing a safe extraction of a single FE card. Fabric plane redundancy is required in this case. In order to stop/start sending traffic over a FAP link use the following API: bcm_fabric_link_control_set(unit, link, bcmFabricLinkTxTrafficDisable, disable); The isolate sequence should disable to links (across all the FAPs) that connected to the relevant fabric plane.
SDK-58641	799444	56640_A0 56640_B0	Added support in SDK software to use NL 11K series 20M external TCAM with the new 56545K SKU.
SDK-58642	799204	56850_A2	Issue :- Trident2 :Action bcmFieldActionIngressGportSet in VFP configures FIELDS_ACTIONf which is not required. Fix :- Added relevant checks to not configure FIELDS_ACTIONf for bcmFieldActionIngressGportSet in Trident2.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-58653		56649_A0 56643_B0 56648_B0 56640_B0 56854_B0 56644_B0 56854_A0 56649_B0 56644_A1 56643_A1 56640_A1 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1 56640_A0 56643_A0 56644_A0 56641_A0 56642_A0 56850_A0 56855_A0 56645_A0 56648_A0 56850_A1 56855_A2 56852_A2 56853_A2 56854_A2 56851P_A2 56851_A2 56850_A2 56851_A1 56851P_A1	In previous release, there was no support to configure RTAG7_MPLS_L2_PAYLOAD_L3_HASH_FIELD_BMAP.MPLS_L2_PAYLOAD_L3_BITMAP_A/B, RTAG7_HASH_CONTROL_2_64.MPLS_PAYLOAD_HASH_SELECT_A/B on KATANA2. This has been resolved by providing and implementing bcmSwitchHashL3L2MPLSField0/1, bcmSwitchHashL2MPLSPayloadSelect0/1.
SDK-58698	799433	56150_A0 53344_A0	An unexpected timeout of draining packets happened when unplugging the cable from TSC ports of BCM53344 platform. This has been resolved with correct port disable sequence.
SDK-58699	795957	56640_B0	Although flexible scaling feature was supported in TR3, due to a bug in TR3 feature list, flexible scaling feature for LPM was not enabled. Applications were not able to add 128b routes due to this. Enabled flexible scaling feature in TR3.
SDK-58725	799988	56450_A0 56450_B0	In the previous releases when more than one MPLS port share the same egress queue and if one of the MPLS port sharing the queue was removed then ING_QUEUE_MAP was deleted as no reference counter was maintained on ING_QUEUE_MAP. From this release added support for maintenance of reference counter on ING_QUEUE_MAP entries. ING_QUEUE_MAP entry is deleted only when all MPLS ports sharing the egress queue entry are deleted.
SDK-58749 SDK-63028	800086	56450_A0 56450_B0	1G speed is advertised with XE port or <=10000 only and not on HIGIG port After flex-io operation on hg port, max-speed still remains as hi gig speed (>=13g) and port is not treated as normal ethernet (ge/xe port). Speed Capability decision was based on max-speed. Due to this max- hi-gig speed, changing speed to 1G port was not possible. Now ORed max-speed with IS_XE_PORT() and issue is resolved
SDK-58750	792732	88650_A0 88650_B0 88650_B1 88660_A0	Trill Adjacency check: 1. encap_id (Link-layer-outlif) now support full range 2. Fix information of bcm_trill_port_get for Adj. database.
SDK-58848 SDK-58983	801222	All	Fixed major lock release issues in SDK code.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-58855	778638	56850_A1	The test in this JIRA involved incomplete AN mode for a given port and that port was flex-ported to another port mode. Sometime HW requires a clean up to clear the incomplete AN mode (CRTSC_616). This JIRA fix adds the fix to clean up the incomplete AN mode prior to flex port transition.
SDK-58856	795594	All	The CLI command "dump pcic" has been made independent of the BDE functions and instead it would now use the cm_device_t function vectors
SDK-58886	802162	All 56334_B0 56334_A0 56850_A0 56850_A1 56850_A2	In earlier releases, internal resource (next_hop/vx_and_swap objects etc.) could not be deleted if customer invoked bcmx_trunk_destroy first and then bcmx_mpls_port_delete. This has been fixed in this release.
SDK-58890	781333	56440_A0 56440_A1	For Katana (and Saber) the counter SOC_COUNTER_NON_DMA_COSQ_DROP_PKT is implemented with a flex counter which counts packets that are discarded by the MMU. By default, this counts packets that are discarded by the MMU for a variety of reasons. One of them is oversized packets. When the MMU flushes a packet because it is oversized, in increments this flex counter. This is not the desired behavior. The fix is to set the bit OP_CNT_CFG in THDO_DROP_CTR_CONFIG which would ensure that counters will only count packets that have been dropped due to output thresholding checks, i.e. those within op_drop_mask.
SDK-58895		88650_B1 88660_A0	This table is static table. Therefore shadowing this memory should be allowed. This table removed from dynamic table list.
SDK-58901	799623	88650_A0 88660_A0	When L2 frame has an 802.3 format where the Ethertype is a Length field, the EXP/DSCP value does not match TC/DP to EXP/DSCP mapping table. This happens when packet is bridged and encapsulated by PWE or IP tunnels. The above functionality conflicts with ECN capabilities. In case ECN is not required in the system we introduce two new SOC properties to disable ECN functionality : mpls_ecn_mode value 0-disabled, ip_ecn_mode value 0-disabled. In case ECN is required in the system we suggest workaround in FP cint_field_ecn_cni_extract.c
SDK-58906		88670_A0	Jericho Global LIF: Support the Jericho LIF differentiation between a Global LIF and a Local LIF.
SDK-58913		56850_A0 56850_A1 56850_A2	In the previous release, the SVP multiple MTPs mirroring and DVP multiple MTPs mirroring were not supported. In this release, this issue has been addressed by adding code changes to support these two mirroring modes.
SDK-58919 SDK-58191 SDK-58894	799048	All	BCM diag shell commands did not work for a system in which the total number of devices <= device ID (i.e. when they are 'holes' in unit IDs, for example - 2 units, IDs '0' & '2').

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-58920	802188	56450_B0	Earlier MSI EP mapping was implemented within the BDE (that supported MSI interrupts) .Customers who did not plan to use the Broadcom BDE then had to implement the MSI mapping in their BDE code. This issue has been fixed by adding the MSI EP mapping to SOC function 'soc_pci_ep_config'.
SDK-58958	802230	All	Added check for XW ports used in helix4 while enabling Frame length check.
SDK-58969	797443	56850_A0 56850_A1 56850_A2	In previous releases, the macro definition MAC_LSB_SHIFT was different for big endian and little endian system. Actually it should be same in both endian systems. This has been fixed in this release.
SDK-58981	798049	56850_A0 56850_A1 56850_A2	In earlier releases SDK destination port was modified unexpectedly when aged through bcm_l2_replace. This has been resolved.
SDK-59047	802819	56450_B0	In BCM 5645X chips after flex port operation the packets will not egress if the port uses external memory. Resolved by adding the missing port reinit for mmu after flex port operation.
SDK-59048	803885	56450_B0	Description: Unexpected Parity Error on LLS_ERROR register. The root cause of this issue is as following: When changing port loopback mode from LB=MAC or LB=PHY to LB=NONE, the operated port will link down after a while and in the time gap, traffic won't be stopped forwarding to the operated port until link down happens to it. Meanwhile, the mode changing squences will modify the MMU associated to the operated port, this causes the MMU issue. Solution: When changing port loop back mode from LB=MAC or LB=PHY to LB=NONE, adding code to force the port to link down to prevent traffic being fordwarded to the port which is being operated.
SDK-59058	790799	56640_A0 56640_A1 56640_B0	This is an improvement to clear initialize the external TCAM (NL11K) that can be used in conjunction with 56640 chips. The entire TCAM database is cleared using a fast initialization sequence in order to keep the time consumed to a minimum.
SDK-59066	793397	56850_A0	In previous releases, the customer could experience a segment error when using stacking application sample code to configure symmetric HG stack links on TD2. This was caused by the SDK using an incorrect maximum number of fabric trunk groups. The issue has been fixed.

Table 100:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.2***
SDK-59074	800863	`56846_A0`	SER injection is accomplished by disabling parity, injecting error into hardware table, and then enabling parity again. In earlier SDK releases, `soc_trident_pipe_select()` which is used to set pipe-line on Trident was invoked after disabling parity. So the parity control register on Y-pipe was not configured and parity was not disabled on Y-pipe. In this release, `soc_trident_pipe_select()` has been placed before setting parity control register to disable parity.
SDK-59087	804148	`56440_A0 56450_A0 56450_B0`	In the previous release the `STRICT_PREAMBLE` and `PROCESS_VARIABLE_PREAMBLE` was not configured for speed less than 2.5G. This issue has now been addressed by setting `STRICT_PREAMBLE=0` and `PROCESS_VARIABLE_PREAMBLE=1` in `XMAC_RX_CTRL` for speed less than and equal to 2.5G
SDK-59101	790579	All	Fixed Linux kernel link errors appeared when Makefile did not include DNX chips.
SDK-59154		`56850_A0 56850_A1 56850_A2`	In previous releases, in `bcm_l3_route_add` API if parameter flags and L3 interface object consistency were not checked for ECMP and could lead to wrong reference count update. In current release, parameter flags and L3 interface object consistency are checked for ECMP.
SDK-59155		`88670_A0`	Added SOC properties for configuring core symmetrically, asymmetrically, and number of active cores.
SDK-59178	803680	`88750_B0 88650_B1`	This issue fixed at 6.3.7 and 6.4.2 versions. The fix is solving a problem which produced at SDK-58398 (masking all interrupts by default).
SDK-59194	804732	All	In the previous release, the left egress mirror could not work if one of the two egress mirrors was deleted when flexible mirroring destinations was enabled. In this release, this issue has been addressed by programming the register `MIRROR_SELECT` correctly when the egress mirror is deleted.
SDK-59198	800106	`56846_A0 56840_A0`	In previous release, "PORT" parameter was not retrieved correctly for registers per warpcore in the routines `_soc_trident_parity_reg_set` and `_soc_trident_parity_reg_get` on Trident+ devices. This would cause system crash when parity error occurred in these registers. The operation that calculates and gets the correct PORT parameter for registers per warpcore has been added for these two routines.
SDK-59211		`88650_A0 88650_B0 88660_A0`	MPLS: when using `bcm_mpls_port_get` field `vccv_type` was not filled (only value supported is bcmMplsPortControlChannelTtl). This is fixed.
SDK-59219		`88670_A0`	Support 2nd myMAC function in Jericho for ROO improvements.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-59235		56640_A0 56640_A1 56640_B0	In previous releases, when trying to create an egress vlan entry on BCM5664X, for virtual ports, the API was giving an assertion, because it was trying to configure an invalid field in egress vlan xlate memory. This is fixed by correcting the field name.
SDK-59237		56334_A0 56304_B1 56150_A0	APIs required for supporting 1-step and 2-step timestamping for Hurricane2 devices has been added in this release.
SDK-59238		56450_A0	KT2 1588 Transparent timestamping support is added.
SDK-59250	770333	56640_A0 56640_B0	In earlier releases PFC on 100G port was not working correctly. When configuring HSP port on TR3 there is a 1 bit shift seen in cos mapping. Priority 2 is mapped to cos0, etc. This is because UC COS0 is tied to L0.1 and so on. This was fixed to assign L0 nodes such that first L0 node is 4X-1. HSP port consumes only 2 index of FC_MAP_TBL since it looks like the mapping in Tr3 would be similar to TD2 for HSP case. Thus the hw_index is changed to 236+8 for ce port 1.
SDK-59252		88650_B1 88660_A0	In FCoE module, when updating NPV port control, using API bcm_port_control_set(unit, src_port, bcmPortControlFcoeNetworkPort, VALUE) behavior is changed to be consistent with other devices flow. Now, to set port to source routing, VALUE should be set to '0'. For port to destination routing, VALUE should be set to '1'.
SDK-59263		56850_A0 56850_A1 56850_A2	Any read operations from CPU could result in the HW aging or L2_BULK triggered HW aging mechanism sometimes incorrectly deleting entries from the L2_ENTRY table. The SW triggered HW aging through L2_BULK operations was introduced to fix this issue. In order to protect L2_BULK operation from all read operations from CPU, not only will the SW aging thread grab the mutex locks for all the memory read operations needed before the L2_BULK operations to trigger the HW aging, but also has the highest priority in all SAL threads to void preemption by other SDK threads.
SDK-59266	806043	56846_A1	In previous releases the sizes of l2_hitda_only and l2_hitsa_only tables were 16K. The fix was to reduce l2_hitda_only and l2_hitsa_only tables to 8K, as each of them can contain the hit bits for 8 L2_entry and total L2_entry is 64K.
SDK-59286	804564	56629_B0 56624_A0	BCM_COLOR_INNER_CFI color setting for "bcm_port_tpid_add" API is not valid for TRX devices. TRX only allows OUTER_CFI to CNG mapping. But the API was not returning error. This has been fixed and will now reject BCM_COLOR_INNER_CFI color setting for "bcm_port_tpid_add" and return BCM_E_UNAVAIL.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-59290	804355	56640_A0 56640_A1 56640_B0	IPv6 Packets were not hitting the external FP rule that matches SMAC and VRF for _FP_EXT_ACL_L2_IPV6 mode. The issue is fixed by changing the offsets of qualifiers during external field qualifier initialization and by programming the correct offsets in LTR (Logical Table Registers) .
SDK-59307	804642	All 56450_A0 56450_B0	In the previous release the tr 140 test did not have an erring pattern print when BIST fails. The erring pattern print has been added now to locate the exact memory location it fails and the pattern used only when BIST fails.
SDK-59327		56340_A0	bcm_regex_match_check() was not taking account of chaining overhead while calculating regex DFA size, This issue has been fixed.
SDK-59337		88750_A0	serdes_qrtt_active soc property has 2 bugs: - it is unable to use it without .<unit number> - it has a bug if the quad number isn't divisible by 4 serdes_qrtt_active soc property is fully supported, and can be used without .<unit> and with any valid quad number
SDK-59352		56540_A0 56540_B0	On FB4 and TR3, the field name SUPPRESS_VXLT in FP_POLICY_TABLE was changed to HI_PRI_SUPPRESS_VXLT. And hence the SDK crashed due to accessing the invalid field during entry install. Fixed the issue by validating the field before accessing.
SDK-59358	805970	88650_A0 88660_A0	Fix to Arad ingress reset sequence
SDK-59371		56640_A0 56340_A0 56850_A2	Problem: No qualifier to match on class value assigned based on SourceGport in Egress Stage of FieldProcessor. Solution: A new qualifier (IngressInterfaceClassVPort) is implemented for matching Source GPort based class value in keys defined for bcmstageEgress. This can be used to match a value assigned to a packet using bcm_port_class_set based on ingress Gport / source Gport in Egress Stage of Field Processor.
SDK-59383	806046	56850_A0	In previous releases, soc_mem_generic_insert is running infinitely doing ser correction and soc_schan_op for the table L3_ENTRY_IPV4_UNICAST if there is parity error in this table. For some reason if the correction does not succeed even though soc_ser_sram_correction returns SOC_E_NONE, then soc_mem_generic_insert does SER correction infinitely. A schan retry limit has been introduced. Once this limit has been reached, ser correction and table operation retry will be terminated and the infinite loop will be avoided.
SDK-59405	806511	88030_B0	Added code to support SGMII ports.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-59408 SDK-43524	803517	56840_A0	WC40 port stopped egressing pkts after link down/up event because of errata 2.6 mentioned in BCM56840-ES107. SW work around is provided to reset CL82 state machine when port goes down.
SDK-59414		56440_A0 56440_B0	On BCM56440, the default TSPLL reference has been changed from an external 12.8MHz reference to the internal TSPLL source. Users with an external TSPLL reference should set the configuration SOC property `PTP_TS_PLL_FREF` to the value of the external reference frequency in Hz. For a 12.8MHz oscillator, `PTP_TS_PLL_FREF=12800000` should be used.
SDK-59417		56850_A0 56850_A1 56850_A2	In earlier release, when trying to clear the dest hit bit in L2 entry, L2 destination was unexpectedly replaced by `bcm_l2_replace()` API. This has been resolved.
SDK-59419		88660_A0	IHB 1B & 2B ECC mask enabled for all memories at Arad+.
SDK-59424		88670_A0	Routing over AC: Provided a CINT example for the application - `cint_route_over_ac.c`. The CINT should be uses only by the Jericho device.
SDK-59425		88650_A0	In TCAM management, TCAM banks may be accessed by different block 'owners': Vlan Translation (VT), Tunnel Termination (TT), Forwarding, OAM, Ingress-PMF-1st-cycle, Ingress-PMF-2nd-cycle, Egress-PMF. A single owner can be configured per bank so that two blocks are not accessing the bank at the same clock. This limitation was not prevented in the current Drivers for specifing owners (e.g. when TCAM bank owner is VTT, then it can be accessed by VT and TT, same for Ingress-PMF Cycle-0 and Cycle-1). This is fixed.
SDK-59455	806478	56450_A0 56450_B0	When L3 ingress mode is set, added support in `bcm_mpls_tunnel_switch_get()` API to return `info->ingress_if` for BCM5644x and BCM5645x devices.
SDK-59464	791410	88660_A0	Cosq: Two flags are added as below: 1) `BCM_COSQ_FC_PORT_OVER_PFC` for Mapping PFC source to port target. 2) `BCM_COSQ_FC_INTF_COSQ_PFC` for Mapping PFC source to relevant priority in all ports on same interface. For a calling sequence example, refer to `cint_arad_pfc_reception_mapping.c`.
SDK-59476		88660_A0	The avs read diagnostics shell command was broken for Arad+ (worked properly for Arad). This command is used to retrieve recommended operating voltage as define upon manufacturing.
SDK-59495		56340_A0	Improvement was requested to add FLOW end reason code in App-IQ engine generated Flow Tracker end report. Provided the support for the same.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-59527	806930	56450_A0 56450_B0	Enabled bcm_port_l3_mtu_set/_get APIs for BCM5645x and BCM5644x devices.
SDK-59529	807602	56450_A0 56450_B0	Support has been added for CoE subport gport in bcm_port_tgid_set/get API for BCM5645x devices.
SDK-59548		56640_A0 56450_A0 56640_A1 56640_B0 56450_B0	In earlier releases there was no SDK support for inner VLAN tag modification. Now fixed to add the support for inner VLAN tag modification through the usage of bcm_qos_map_XXX( ) API and BCM_QOS_MAP_L2_INNER_TAG flag to create profiles in EGR_MPLS_EXP_MAPPING_2 table.
SDK-59553		88650_A0 88660_A0	OAM: bcm_oam_loss_get() returned incorrect values because loss_farend and loss_nearend were miscalculated.
SDK-59568		All	In earlier releases, there was no way to use the API bcm_rate_bandwidth_set to set the rate to 0kbps speed. If the speed was set to zero, the assumption was to disable metering. Now, to disable metering, kbits_sec parameter has to be passed as BCM_RATE_DISABLE and to set the rate to zero, kbits_sec parameter has to be passed as BCM_RATE_BLOCK
SDK-59570		All	In earlier releases, there is no way to use the API bcm_rate_bandwidth_set to set the rate to 0kbps speed. If the pass the speed as zero, the assumption was to disable metering. Now, to disable metering, kbits_sec parameter has to be passed as BCM_RATE_DISABLE and to set the rate to zero, kbits_sec parameter has to be passed as BCM_RATE_BLOCK.
SDK-59573	806883	88030_B0	handle TMU error interrupt to avoid hash collision errors during hash insert for bcm88030
SDK-59590	798399	88650_B1	Cosq: CFC_EN bit in CFC_ENABLERS register is used to enable device-level flow control. This bit should be set after all other CFC settings are done. Otherwise, some errors may be occur. After the fix, the bit will be set after all other CFC settings are finished.
SDK-59617		56450_B0	Added support to use port class id upto 127 in bcm_vlan_translate_egress_action_add() API for BCM5645x devices.
SDK-59636		88650_A0 88660_A0 88670_A0	Add a new class, bcmPortClassProgramEditorEgressPacketProcessing , for the bcm_port_class_get/set API. Used to manage a programmable editor variable in the egress.
SDK-59669	808665	56450_A0 56450_B0	In the previous releases the TTL of L3 route label cannot be read back via bcm_l3_egress_get. This has been fixed to read TTL value of L3 route label for requested entry from EGR_MPLS_VC_AND_SWAP_LABEL_TABLE table and return to the user via bcm_l3_egress_get.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-59681		88650_A0 88660_A0 88670_A0	In order to prevent redundant printouts upon bcm.user/pcid initialization, following need to be added to soc file : debug bcm link,attach warn debug soc ddr warn debug soc common err debug sys verinet warn
SDK-59686		88650_A0 88650_B0 88650_B1 88660_A0 88670_A0	In Ingress Field Processor, using qualifiers bcmFieldQualifyIp4/Ip6/Mpls indicates that bcmFieldQualifyInnerSrcMac and bcmFieldQualifyInnerDstMacqualifiers should be taken from third sub-header instead of the second. In practice, when bcmFieldQualifyIp4/Ip6/Mpls are used, then all qualifiers are taken from wrong header, instead of only inner Source/Destionation Mac. This is fixed.
SDK-59690	804106	56340_A0	In this release added the phy control bcm shell comand support for qsgmii serdes of Helix4
SDK-59700		56850_A0 56850_A1 56850_A2	Description: SER failed to correct entry of ING_FLEX?CTR_COUNTER_TABLE_[0 to 7] once a SER error occurred on it. Root cause: The reported entry index is the offset of the reported entry to the first entry of ING_FLEX_CTR_COUNTER_TABLE_0, but it should be the offset of the reported entry to the first entry of ING_FLEX_CTR_COUNTER_TABLE_X which the entry belongs to. Solution: The software now calculates the index again once we discover the reported entry is one of ING_FLEX_CTR_COUNTER_TABLE_[0 - 7], the algorithm is using lower 12 bits of the index as the new index.
SDK-59705	805285	All	In L2 MAC table, when calling the API function bcm_l2_addr_delete_by_trunk(), the trunk ID was not masked properly when traversing the MACT, resulting in unintended deletion of entries. The masking of the trunk ID is now fixed.
SDK-59710		88650_B1 88660_A0	AC gport can be found via matching info. An error occurs when trying to get LIF gport via matching info for ingress only ACs. After the fix, VLAN gport can be found with BCM_VLAN_PORT_CREATE_INGRESS_ONLY flag.
SDK-59761	808170	56450_A0 56455_A0 56450_B0	Port command was not working properly in SDK for the Port advertisement field. The port advert mode was not getting set when done via CLI command. Root cause of this issue was that the action mask that enables the port advert field was altered to default this is port ability instead of port mode. Handled this issue by modifying the action mask that enables port advert field to set it by default to port mode.
SDK-59764		88660_A0	BFD: properly set the BFD rate given the parameter bfd_period in bcm_bfd_endpoint_create().

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-59823	808861	56850_A0 84328_A0 56850_A1 84328_B0 56850_A2	It was reported that when there is an external PHY that doesn't support TX squelch, the squelching action would return "SOC_E_UNAVAIL", which would lead to the failures of some loopback related operations. The fixing was to use PHY notify event to squelch the TX on internal PHY instead of calling PHY control set to squelch the TX on external PHY when the external PHY is present.
SDK-59833	804363	56850_A0 56850_A1 56850_A2	In previous releases, the conflicting MAC addresses couldn't be retrieved correctly due to a typo in _bcm_tr_dual_l2_conflict_get. The issue had been fixed.
SDK-59839		88650_A0 88660_A0	L3: The API bcm_l3_ingress_get() retrieve some RIF related info that can also be set through the related API bcm_l3_ingress_create(). One of the retrieved fields is the VRF of the RIF. The VRF field wasn't updated by bcm_l3_ingress_get() although it was set by bcm_l3_ingress_create(). bcm_l3_ingress_get() was modified to retrieve the VRF field.
SDK-59853		88660_A0	In ARAD+, when using external TCAM for forwarding, RPF and forwarding searches were always performed serially on a single database, using SIP and DIP respectively in search keys. A new SOC property is added to enable the user to perform forwarding and RPF searches on duplicated databases in parallel (as in ARAD): - custom_feature_ext_rpf_fwd_parallel - when set, forwarding and RPF searches are performed in parallel. When the SOC property is set, in case of IPv4/6 + RPF forwarding query, external ACL databases IDs are changed to 2 and 3. The actions sizes for ACL databases are also changed accordingly: The action size for ACL database 0 is 64 bits. The action size for ACL database 1 is 16 bits. The action size for ACL database 2 is 32 bits. The action size for ACL database 3 is 24 bits. All of the changes above apply only to ARAD+ devices.
SDK-59854	809432	56340_A0 56547_A0	wcmod_power_control() was powering off global PLL block when one lane was disabled, which caused all 4 lanes to be powered down. Fixed to power down global PLL block when all 4 lanes are disabled.
SDK-59862	779893	88650_B1 88660_A0	Stacking systems are transmitting between the different systems the whole packet, including the internal headers (FTMH + extensions, PPH + extensions, User-Headers if present). In case the user-headers are present in the system, the second stacking system was adding again user-headers to the packet. This is fixed.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-59864 SDK-61165 SDK-61022		88660_A0	In metering, the meter configuration is optimized for the best rate accuracy. However the configuration was not optimized to have the highest bucket update rate. This created some issues when the traffic is composed of bursts and the bucket size is small. This fix changes the calculation of the values to configure to have the best possible bucket update rate, without having any impact on rate accuracy.
SDK-59866		56450_A0 56450_B0	Problem: Access to ING_PHYSICAL_PORT_TABLE in traffic was causing corruption and behaviour was un-predictable. Fix: 1) Added KT2-B1 support 2) Fix for ING_PHYSICAL_PORT_TABLE corruption happening in hw Read and write to ING_PHYSICAL_PORT_TABLE access is protected by writing 1 to 0 INQ_Q_BEGINr:ING_PHYSICAL_PORT_SBUS_WITH_PKT_DISABLEf
SDK-59898	701984	88650_A0	Disabling a port, using bcm_port_enable_set() fails since EGQ fail to drain. Fixed
SDK-59917		56640_A0 56640_A1 56640_B0	bcm_cosq_gport_scheduler_set() API does not seem to take effect while setting the scheduler to WRR. Turns out, even though the API does set the register for enabling WRR initially, it gets overwritten later on. This has been rectified by changing code such that the register change is done towards the end of the API implementation.
SDK-59934	811343	56850_A0	Root cause: Once a ser error occured on a entry of memory of L3_DEFIPm, the correction handle will invoke routine soc_generic_ser_process_error to correct it. This routine is a common one for all chips. But each chip inputs a chip based specific event not a BCM event. The chip based specific event varies on each chip. But in the routine, it only deals a SER error, so we need raise a PARITY error detected event. The existing code really has the code to raise a event but with a wrong type since it used the err_type directly. As a result, the event reported in this case would be different. After invoking the SER correction routine, a SOC_SWITCH_EVENT_DATA_ERROR_LOG should be reported if there is a valid log ID avaiable. But the existing code did nothing. Solution: Correct the wrong reported type to SOC_SWITCH_EVENT_DATA_ERROR_PARITY and adding code to report SOC_SWITCH_EVENT_DATA_ERROR_LOG if a valid log ID is avaiable after invoking the SER correction routine can resolve the issue.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-59955	808817	88650_A0 88650_B0 88650_B1 88660_A0	L2 MACT Flush: Added support for 2 fields flush in application-based-allocation mode using bcm_l2_replace_match api with valid match_addr.dest & match_addr.encap_id fields.
SDK-59966		56640_B0 56850_A2	Fixed an indexing issue in LPM algorithm that allocates space for the routes in L3_DEFIP TCAM. The issue was specific to the case of the TCAM being in paired mode and LPM scaling is enabled (config property lpm_scaling_enable=1).
SDK-59974	809811	56440_A0 56440_A1 56450_A0 56440_B0 56455_A0 56450_B0	In the previous release the priority given in bcm_vlan_dtag_add was not taken as the outer tag priority. ING_VLAN_TAG_ACTION_PROFILE table needed to program by setting SIT_OPRI_ACTION field to set the priority field.
SDK-60017	806360	88650_A0 88640_A0 88650_B0 88650_B1	OAM: Reduce running time of bcm_oam_endpoint_create() and bcm_oam_endpoint_action_set() by 50%-60%
SDK-60025	786758	88650_A0 88650ACP_A0 88650_B0 88650_B1 88660_A0	In Counter Processor, under Egress TM mode, the Counter Engine can differentiate between different Traffic Class, even if the port is set under one priority level. This was blocked in the Driver, and fixed now.
SDK-60029		88650_A0	Cell drops occurred on a system with single fabric pipe BCM88650 and dual fabric pipe BCM88750. Such a single fabric pipe BCM88650 should be configured as follow: 1. Set to single fabric pipe mode is_dual_mode.BCM88650=0 2. Adjust fabric cell format to dual pipe mode system system_is_dual_mode_in_system.BCM88650=1
SDK-60041	812854	56150_A0	Fix the packet loss issue during EEE mode on GE port with UNIMAC.
SDK-60042		88660_A0	Added support in BFD Remote Server. To add BFD endpoint to the OAMP server the flag BCM_BFD_ENDPOINT_HW_ACCELERATION_SET should be used in bcm_bfd_endpoint_create api. For usage example see cint_bfd.c
SDK-60048	803308	56624_B0 56624_A0 56624_A1 56514_A0 56334_A0	memscan support is added for TCAM error detection/correction on 56624, 56334 and 56514 chips.
SDK-60049		88650_A0 88650_B0 88650_B1 88660_A0	soc property length limitation cause segmentation fault at trill init. Change soc property max length to 128 characters and add test to prevent segmentation fault.
SDK-60056	812600	88650_B1 88660_A0 88670_A0	In Field Processor, the support of cascaded ingress-egress lookup was possible only for packets without Learn-Extension header. If custom_feature_cascaded egress learning SOC property is enabled, a Learn-Extension header is added to all the Ethernet packets, and cascaded ingress-egress lookup is supported in this case.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60070	811441	56450_A0 56450_B0	Problem: Earlier interface mode retrieval method was based on PortType in legacy Xenia driver code (i.e. XE_TYPE ==> XGMII, GE ==>GMII etc). For WC core driver, interface mode was based on port speed. This was creating confusion Solution: Chaged interface mode retrieval method in Xenia driver. Now it will be based on port speed (i.e. 10G ==> XGMII, 1000/2500G == GMII , 100M == MII) This will make behaviour same for both xenia/ unicore and warp-core driver and no confusion should appear now.
SDK-60072	812645	88650_B1	An HW bug is present in EGQ block when terminating more than 64 Bytes (see errata document for more information). A work-around was proposed in srcexamplesdppcint_field_ingress_large_termination.c, where 14 Bytes were terminated at ingress if the Forwarding-Header was located more than 32 bytes away from start of packet. In certain conditions (e.g., in stacking systems with multiple VLAN tags), removing 14 Bytes was not sufficient. It has been fixed to 28 Bytes.
SDK-60073	794596	All 56850_A0 56850_A1 56850_A2	Add two flags for extender usage: BCM_EXTENDER_PORT_DROP and BCM_EXTENDER_PORT_ID_ASSIGN_DISABLE. When BCM_EXTENDER_PORT_DROP is set, packets egress to the indicated extender port are dropped. When BCM_EXTENDER_PORT_ID_ASSIGN_DISABLE is set, matched packets are not forwarded to the extender port.
SDK-60085	810017	56640_A0	On Triumph_3, when parity error was induced by disabling CMIC_SER0_PROTECT_ADDR_RANGE_VALID and subsequently detected in SBUS DMA by Memscan, no action was being taken to recover from this error. This was because when memscan encounters DMA failure and falls back to individual memory read, data was being read from the cache and not from h/w and hence the parity error was not being caught. Corrected this to read from h/w which issues an Schan operation which then recovers from parity error.
SDK-60094	810331	88660_A0	Routing Over VXLAN in ARAD+: When bcm_l2_egress_create is called with different SMAC, the encapsulation of outer SMAC is always as the first allocated SMAC.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60099		56850_A0 56850_A1 56850_A2	Issue: bcm_field_qualify_IpType is setting right h/w encoding for bcmFieldIpTypeNonIp but bcm_field_qualify_IpType_get is failing due to wrong h/w encoding value derived in internal get function. Fix: modified internal get/set functions to match h/w encoding for bcmFieldIpTypeNonIp.
SDK-60102		56450_A0 56440_A1 56450_B0 56440_A0 56455_A0 56440_B0	In the previous release the priority given in bcm_vlan_dtag_add was not taken as the outer tag priority. ING_VLAN_TAG_ACTION_PROFILE table needed to program by setting SIT_OPRI_ACTION field to set the priority field.
SDK-60109		56340_A0	_bcm_tr3_flex_ctr_pool_unmap was not releasing all reserved flex counters by _bcm_tr3_flex_ctr_pool_map. Now fixed _bcm_tr3_flex_ctr_pool_unmap to release all reserved flex counters for Appiq.
SDK-60110		88650_A0 88650_B0 88650_B1	For ISSU, in warmboot-engine module, an option is given internally to skip a variable when retrieving the external storage in a newer version. This feature had issues discovered in its first usage in 6.3.8: the Driver is still trying to access these variables in some part of the code, which may lead to errors in the variable offset computation. This is fixed.
SDK-60131		56334_B0 56334_A0	Problem : When you create a child and parent policer in a slice, it takes up 2 complete meter pools and reserves them completely for all parent and child policers created in that slice. And those meter pools will not be shared by any policers from other slices. Because of this, we are prematurely exhausting policer resources. Requirement here is to use same METER POOL for policers created in different slices. Since Level0 policers of a slice(i.e. group) cannot be shared by other slices, we need to expand level-1 meter creation logic to use same meter pool for different slices. Solution: Adding new variable to bcm_policer_config_t structure to save the meter pool id. User can retrieve the meter pool_id and create a policer with give meter pool id so that the meters are created in the same pool. But the field entries related to meters in same pool need to be mutually exclusive. Added required doc changes for above implementation.
SDK-60144	808055	88650_A0 88660_A0	added a lock in device reset so other threads won't access the device
SDK-60166	808650	56850_A0 56340_A0 56850_A1 56850_A2	In previous releases it was reported that bcm_vlan_translate_egress_action_add etc. APIs would return BCM_E_PORT error when passed a VLAN port type gport as an input parameter. This issue has been fixed by adding the VLAN type gport as a condition during gport type check so that not to return BCM_E_PORT directly.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60168	813722	All 56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In earlier releases, vlan port vp could not be destroyed when associated gport was a trunk and this trunk did not have members. This has been resolved.
SDK-60169		56850_A0 56850_A1 56850_A2	In the previous release, when two SVPs created on the same physical port were configured with two mirroring sessions pointing to different MTPs, the MTP of the first mirroring session would be overwritten by the second MTP, which was incorrect. In this release, this issue has been addressed by adding a new bookkeeping to ensure two MTP slots can be allocated in the case.
SDK-60170	813670	All	When the external phy is in repeater mode, then HW linkscan for those ports should scan the internal serdes link status instead of external phy's. In order to acheive this, the repeater flag was set for BCM84756. Linkscan will only scan the internal serdes link status if the flag is true.
SDK-60179	810107	56540_A0	64bit index space was not initialized in case of rpf mode and 128b entries disabled mode. Due to this applications were not able to add routes in this particular configuration.The fix is to initialized the 64 bit index space to right values.
SDK-60194	813706	56850_A0 56850_A1 56850_A2	In this release added support for BCM_PORT_PHY_CONTROL_RX_SIGNAL_DETECT APIs (get and set) to TSCMOD
SDK-60199	812394	56450_A0 56455_A0	In prior releases both OPRI_CFI_SELf and IPRI_CFI_SELf fields in EGR_VLAN_XLATE table were set to "1" if the priority is set to "-1" while adding entry to EGR_VLAN_XLATE table though the action configured includes only for outer tag. This impacted LSP EXP bit remapping not to work. Now fixed to set IPRI_CFI_SELf action includes configuration for inner tag and OPRI_CFI_SELf if the action includes configuration for outer tag.

Table 100:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.2*
SDK-60222 SDK-60187		88650_A0 88650_B0 88660_A0	added memory array caching: Some memories have multiple instances within a single block in the device. Internally, the SDK organizes such tables into arrays, so that each instance of a table can be accessed using an instance-id. The caching mechanism did not support caching of memory-arrays. Customer impact: Normally, memories can be cached to DRAM as part of soft-error (SER) protection mechanism, or in order to improve driver runtime performance. For memory arrays, caching was not available. For parity-protected configuration memories, caching is used to recover from a soft-error. As a result of this addition, memory arrays can now also be protected by this mechanism. list of relevant memories: EPNI_EEDB_BANK EPNI_DATA_FORMAT EPNI_LFEM_FIELD_SELECT_MAP IDR_MCDA_PRFSEL IDR_MCDB_PRFSEL IHB_FEM_BIT_SELECT IHB_FEM_MAP_INDEX_TABLE IHB_FEC_ENTRY IHB_TCAM_ACTION IHP_PARSER_PROGRAM IHP_VLAN_PORT_MEMBERSHIP_TABLE IHP_LIF_TABLE note for 6.3 branch: the list of cached tables is not filled by the fix in this JIRA, and will be filled soon.
SDK-60225		88650_A0	Reference only: Parity memories are added to the reference shadow list in order to be cached by default.
SDK-60233	815117	56640_A0 56640_A1 56640_B0	In earlier releases, the parity information structures for MMU_MTRO memories were incorrect. Hence the recovery logic fails to correct the error, and hence the error was being reported continuously. This is now fixed by correcting the parity structures.
SDK-60241 SDK-43891	814752	56640_B0	Previously, when we have an hash bucket full, and If we try to add an double wide L2 entry, it fails even if we pass an replace flag. Now support has been added to dynamically replace narrow and wide entries with each other, by passing an replace flag. In such cases, and existing entry will be deleted (two entries may be deleted if the newer entry is a wide entry and existing entries are narrow).
SDK-60243	814299	All	Diag Shell command was not able to qualify DstMultipath as parameter. Added case to handle DstMultipath when entered from diag shell.
SDK-60261	815126	88650_A0 88650_B0 88650_B1 88660_A0	fixed bug: EGQ_DSP_PTR_MAP wasn't correctly updated when removing LAG ports that had the BCM_TRUNK_MEMBER_INGRESS_DISABLE flag.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60262	815031	`88650_B1 88660_A0`	In TCAM memory, there is no mechanism for identifying and fixing SER errors in the TCAM. A new mechanism is added which handles an error upon SER occurrence, identifies the location of the error and according to TCAM bank owner, reproduces the TCAM entry from software database and rewrites it to TCAM memory.
SDK-60267	815566	`88650_A0 88660_A0`	In the Counter Processor module, during the background counter read from DMA (aka algorithmic read), the number of counters asked to be read could exceed the maximum size of the counter cache. This is fixed by adding a validation to the number of counters. This prevents the PCIE bus to crash.
SDK-60278	813007	`56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1`	In previous releases, there was an issue for DLB HG trunks that traffic ingress from port located in Y-pipe would be getting dropped. The issue was fixed by configuring `DLB_HGT_FLOWSET_Y` table during DLB setting.
SDK-60280	815365	`56450_A0 56450_B0`	In `bcm_port_tpid_set` API port resolution logic for vp ports created on coe subport was not implemented. Now implemented port resolution logic for vp ports created over coe subports The fix has been updated for `bcm_port_tpid_get` and `bcm_port_tpid_add` APIs as well.
SDK-60301	815641	`56340_A0`	In previous releases when an warpcore block had both with both ieee and higig encapuslations, changing the encapusulation of xe port to higig caused all the other ports to change to 11G, which resulted in linkdown for all the ports. In such cases we needed to limit the maximum speed of xe ports to 10G which can be done by adding the config variable `port_max_sped` for that particular port . This issue has been resolved.
SDK-60302		`56845_A2`	`PQ_XYZ` registers were giving SCHAN read access time out errors with "dump soc" command. These registers aren't open anymore on TD/TD+. So they are all masked now in SDK .
SDK-60304		`88650_A0`	In mesh system, there was no gport type that addresses fap id 0. New gport type added: bcmCosqGportTypeGlobalFabricMeshDev0
SDK-60307 SDK-60336	814621	`56456_B0 56450_A0 56455_A0 56456_A0 56450_B0`	For BCM5645x devices, when config 10 `(bcm5645x_config=10)` was used, the existing TDM used was not fully flexible. So it caused issues while converting 10G port to 4x2.5GE port. The earlier TDM used for this config has been replaced with a fully flexible TDM to support conversion of 10G port to 4x2.5GE ports without any issue for BCM5645x devices using config 10.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60312	812350	88650_B1 88660_A0 88670_A0	OAM: For remote endpoints, a call to bcm_oam_endpoint_get() returns the flag BCM_OAM_ENDPOINT_FAULT_CCM_TIMEOUT in the field faults in cases there is a CCM timeout. Likewise for local endpoints, each of the flags "BCM_OAM_ENDPOINT2_RDI_FROM_RX_DISABLE" and "BCM_OAM_ENDPOINT2_RDI_FROM_LOC_DISABLE" will be on as outputs of bcm_oam_endpoint_get() if and only if these flags were on when calling bcm_oam_endpoint_create() for the respective endpoint.
SDK-60315	814659	56640_A0 56640_A1 56640_B0	Problem: DstPort qualifier used for qualifying cpu packets was getting qualified for transit multicast packets as well. Solution: Extended DstGport qualfier for all devices, where there is control to set the the type of packets that we want to qualify using d_field/d_type fields in FP_TCAM.
SDK-60321	806640	56456_B0 56450_A0 56455_A0 56456_A0 56450_B0	Added switch control bcmSwitchTimesyncEgressTimestampingMode to choose between 32bit or 48bit timestamping mode for 1588 packets. Also new flag BCM_PORT_TIMESYNC_TIMESTAMP_CFUPDATE_ALL is added to specify that all 1588 packets will be egress timestamped to to the flags field of bcm_port_timesync_config_t struct. Default mode is to timestamp only ingress timestamp packets.
SDK-60336 SDK-60307	813964	56450_A0 56450_B0	For BCM5645x devices, when config 10 (bcm5645x_config=10) was used, the existing TDM used was not fully flexible. So it caused issues while converting 10G port to 4x2.5GE port. The earlier TDM used for this config has been replaced with a fully flexible TDM to support conversion of 10G port to 4x2.5GE port without any issue for BCM5645x devices using config 10.
SDK-60337	816022	56456_B0 56450_A0 56456_A0 56450_B0	1588 protocol's host cpu based applications can use either one-step time-stamping or two-step timestamping mode. In case of,one step timestamping mode, PTP packets correction field is updated with packets transmits timestamp and in two step timestamping mode, PTP packet's egress time-stamp can be retrived from MAC. This JIRA provides support for one-step and two-step timestamping for host CPU based PTP application through bcm_tx() API for Katana 2 chipset.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60353	807644	56846_A0 56840_A0 56850_A0	In previous release, bcm_port_dscp_map_mode_set/get(), bcm_vlan_priority_map_set/get(),bcm_port_dscp_map_mode_set/get() could only work for physical port. Now these APIs have been enhanced to support MPLS PORT on TD, TD+ & TD2. Please note that the GPORT DSCP & DOT1P profiles should be initialized first before being used in these APIs.
SDK-60356		88650_A0 88660_A0	QOS IPV4/MPLS: Added default 1:1 mapping from TC to TOS/EXP in the egress instead of just 0
SDK-60358		88670_A0	OAM: Implementation OAMP-DM trigger mechanism. Refer to user manual for full documentation.
SDK-60368	814626	56850_A0 56850_A1 56850_A2	When BCM_SWITCH_PKT_INFO_HASH_UDP_SOURCE_PORT flag is used, it is supposed to return the udp source port to be used in vxlan packet. So it should fit in 16 bits. Root cause: The underlying routine, compute_td2_rtag7_vxlan() computes the "raw" 20-bit value, which is never masked down to 16 bits for the UDP port option. Solution: bcm_switch_pkt_info_hash_get() is fixed to mask raw 20-bit entropy label value down to 16 bits and return the (64k) value properly.
SDK-60384		88660_A0	Decreasing E2E shaper of a non-active port causes packet drop of another active port within the same channelized interface - Fixed
SDK-60393		88650_A0 88660_A0	In Ingress Field Processor, when creating a Direct Table database, the first Direct table entry may be written to the wrong action table. Per TCAM bank, two action tables can be allocated to the Direct table Field group. This issue happens in case the second table (20b MSBs) is allocated, and no TCAM bank is yet allocated. This is fixed.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60394		88650_A0	Credit scheduler memories are added to the reference-application cache list. Memories can be cached to DRAM as part of soft-error (SER) protection mechanism, or in order to improve driver runtime performance. Upon initialization, caching mechanism usually uses Table DMA to synchronize the HW with the shadowed memory cache. Since the standard table DMA mechanism is not applicable to scheduler tables, initialization for these tables uses non-DMA access. An alternative mechanism is currently being analyzed, and might be available in the future, to improve performance when caching these memories. For parity-protected configuration memories, caching is used to recover from a soft-error. Due to this fix, the following credit scheduler memories added to the cache list and can be protected by this mechanism: a. list of relevant memories: SCH_CIR_SHAPERS_STATIC_TABEL__CSST SCH_CL_SCHEDULERS_CONFIGURATION__SCC SCH_CL_SCHEDULERS_TYPE__SCT SCH_DEVICE_RATE_MEMORY__DRM SCH_DUAL_SHAPER_MEMORY__DSM SCH_FLOW_DESCRIPTOR_MEMORY_STATIC__FDMS SCH_FLOW_GROUP_MEMORY__FGM SCH_FLOW_SUB_FLOW__FSF SCH_FLOW_TO_FIP_MAPPING__FFM SCH_FLOW_TO_QUEUE_MAPPING__FQM SCH_HR_SCHEDULER_CONFIGURATION__SHC SCH_PIR_SHAPERS_STATIC_TABEL__PSST SCH_PORT_SCHEDULER_WEIGHTS__PSW SCH_SCHEDULER_CREDIT_GENERATION_CALENDAR__CAL SCH_SCHEDULER_ENABLE_MEMORY__SEM
SDK-60399		56340_A0	Resetting of logical engine was not detaching logical to physical engine mapping, which caused wrong free size calculation. Fixed it by marking logical to physical engine mapping as NULL after reset.
SDK-60408		56450_A0 56450_B0	Problem: Two consecutive IDLE slots are required in TDM to meet TCAM atomicity. One typo remained in TDM-A2 where instead of IDLE slot value as 63, value remained as 62. It could cause (theoritically) concern for TCAM atomicity in RTL design Also value of AUX_ARB_CONTROL_2r:TCAM_ATOMIC_WRITE_ENABLEf was 0 i.s.o. 1 Fix : Added one more check in tdm verification function for avoiding possibility of IDLE-SLOT=63 in middel of TDM. Also restricted Two consecutive IDLE slots to 63 i.s.o. 62 and 63 Made AUX_ARB_CONTROL_2r:TCAM_ATOMIC_WRITE_ENABLEf value as 1 in KT2:misc() routine

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60434	816207	56450_A0	Deletion of port in range of 120-169 was failing for the multicast group. Fixed the code for the port in range of 120-169 to delete the IPMC_VLAN replication pointer specific to the port for the multicast group, hence the deletion of port from replication list successful.
SDK-60436	816417	88030_A0 88030_B0	When allocate trie node, we need to confirm the valid field of bpm is between [0, skip_len+1] and mask out the higher bits which is invalid.
SDK-60439	814627	56456_B0 56450_A0 56456_A0	Broadcom switch hardware can recover SyncE clocks GE/XE ports. ports. This JIRA adds support to recover SyncE clocks from GE/XE ports for Katana 2 (56450 A0 and 56450 B0) chips.
SDK-60440	817398	88030_A0 88030_B0	Changed the MACROs to ensure they are CALADAN3 specific checks and do not affect any other modules.
SDK-60446	815613	56850_A0 56850_A1 56850_A2	In earlier releases, displaying port bitmaps in the Diag Shell caused a segmentation fault. This has been resolved.
SDK-60464	815760	88650_B1	This memory is not accessible memory for CPU. The fix changes the print and provides correct information.
SDK-60470	817158	56850_A0	In earlier releases, only (<=4k) L3 interfaces per port could be deleted from a multicast group by calling bcm_multicast_egress_delete. This has been fixed in this release.
SDK-60477	813482	56640_B0	added the fix to allow tx transmit during linkdown
SDK-60482	817859	56850_A2	In earlier releases, SDK did not support changing vlan port match condition from vlan port vp to vplag vp when vlan port was added to vplag. This has been resolved.
SDK-60485		88670_A0	Failover: Added a CINT example for 1+1 protection creation in the new Jericho decoupled mode. The sequence can be seen in cint_vswitch_cross_connect_p2p.c upon creation of access side PWE and VLAN ports.
SDK-60486	817588	88650_B1 88660_A0	Reference code: BCM sdk enables some interrupts at the reference code for monitoring and corrective action performance. This fix adds some interrupt to this list in order to enable the monitoring and corrective action: ARAD_INT_EGQ_DBFECC_1BERRINT,// monitoring ARAD_INT_EGQ_DBFECC_2BERRINT,// monitoring ARAD_INT_EGQ_PARITYERRINT,// This interrupt indicates on parity error fault at EGQ parity protected memories. Enabling this interrupt makes parity error corrective action for these memories possible.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60489		88650_A0 88650_B0 88660_A0	Trill MC: In 6.4.X we moved to a new Trill sequence for multicast case. Bug was found when adding MACT L2 entries with multicast group and trill port information. When multicast group is not just a number but also has indication of multicast type (for example when we take multicast_id as a result of bcm_multicast_create) then error is invoked.
SDK-60491	818025	All	Egress True Mirroring destinations were not displayed in diag shell mirror show. Corrected this behavior and added flags to display true egress mirror destinations.
SDK-60494		All	Fixed BCM shell commands prefixed with unit ids indications such as '*:' and '2-5:' for a system in which the total number of devices <= device ID [i.e. when they are 'holes' in unit IDs, for example - 2 units, IDs '0' & '2'] (these fixes where missed in SDK-58919).
SDK-60503	815053	56450_A0	The frame length check was getting enabled always for non Higig ports on BCM5645x devices when doing a flexIO hot-swap operation. This has been fixed by disabling frame length check by default after a flexIO hot-swap operation. The frame length check can be enabled after flexIO hot-swap operation by setting config variable mac_length_check_enable.
SDK-60508	816530	56340_A0 56340M_A0	During multi hash move operation, the destination bucket can have different sized entries than the incoming entry. In earlier releases, the SDK did not care if the incoming entry could fit into destination bucket. Due to this, it could corrupt the existing entry in some cases. The fix is to add an infrastructure to check if incoming entry can be added into the destination bucket. if not, keep looking for next destination bucket until you get a free space or you have reached to max try counter.
SDK-60515	793455	All	Fixed OTN (bypass TDM) in 88650/60 connected by Mesh.
SDK-60527		56846_A0 56850_A2	Problem: After warmboot upgrade, udf field entry was not getting recovered properly. Solution: Added a missing condition while recovering qualifier set for field entries having UDF.

Table 100:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.2*
SDK-60536		56455_A0	In BCM5645x devices, Parity errors were observed during SDK initialization. This was because of parity errors in Channel interfaces were not handled and were also checked for interfaces that were not used. Fixed this issue by merging changes made SDK-35808 from master to 6.3.9 branch via SDK-61032 to modify CI error mask to handle parity errors and also to enable parity checks for only those interfaces supported in device. Also when 156.25Mhz reference clock for LCPLL is enabled in the configuration, system crashes This is because 156.25MHz refclk in LCPLL non-bypass mode is not supported in Katana2 Fixed this issue by removing 156.25Mhz refclk initialization during soc reset.
SDK-60543	811382	88660_A0	Upon accessing link APIs when linkscan is active, the driver might deadlock from time to time on different links. This condition is fixed.
SDK-60549		56150_A0 56151_A0	BroadSync embedded application support added for BCM5615X
SDK-60551		56640_A1	In older releases, for 5664x devices, for Ports to come up as XE ports, the corresponding bits have to be set in the config property pbmp_xport_xe. Now, In addition to XE ports, all the 100G+ (CE) ports have to be added to this config property.
SDK-60560	818014	88650_A0 88650_B0 88650_B1 88660_A0	L2 Flush: Added support for encap_id masking in bcm_l2_replace_match api.
SDK-60576	818773	56450_A0 56450_B0	Problem Statement: Local port resolve is failing when port number is greater than 127 when trying to set the scheduling mode Resolution: Changed the code to use proper function to resolve local port resolve when the GPORT is of type BCM_COSQ_GPORT_SUBSCRIBER
SDK-60583	816905	56340_A0 56547_A0	Parity errors in VLAN_XLATE table detected on packet lookups are reported to ser_fifo as table RAW_ENTRY_TABLE with physical addresses. In previous releases, the index of parity error entry was not obtained correctly in software. In this release this bug has been fixed to enable memory decoding routine to work for RAW_ENTRY_TABLE in Helix_4.
SDK-60590		88650_A0 88650_B1 88660_A0	This JIRA fixes segmentation fault which occurred occasionally at deinit/init sequences. This happened because of improper implementation of interrupt deinit sequence. At this case interrupt asserted during SW interrupt resources deallocation. This fix makes sure that no interrupt will asserted during interrupt application deinit.
SDK-60593	819011	84793_A0	Added per lane eye scan support. Modified enzo (phy84793) driver to return RX sequence done for the specified lane.

Table 100:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.2*
SDK-60595		88650_A0 88660_A0	OAM: For calls to bcm_oam_endpoint_create() with the opcode_flags CCM_OAM_OPCODE_CCM_IN_HW set (accelerated endpoints) calling sequence has been changed: For Up-MEPs the field tx_gport must be set to BCM_GPORT_INVALID. The gport on which OAM PDUs will be transmitted will be determined through the pipeline, as dictated by the protocol for up-MEPs. For Down MEPs the tx_gport field must be set to a valid system port. For Ethernet OAM, two endpoints using the same least significant byte in the src_mac_address field must use the same tx_gport. BFD: Fixed a bug in bcm_bfd_endpoint_create() for accelerated endpoints (calls in which the flags BCM_OAM_OPCODE_CCM_IN_HW is set) causing BFD frames to be sent to an incorrect gport when multiple endpoints exist on different gports.
SDK-60596		88660_A0	OAM: Support transmission/reception of CCMs with TLV where Type=2 (port status TLV) or 4 (interface status TLV). The TLV on transmitted CCMs can be set with the API bcm_oam_endpoint_create(). When setting Type=port status is desired, the flag BCM_OAM_ENDPOINT_PORT_STATE_UPDATE should be set in the flags field and the field port_state should be set to one of the BCM_OAM_PORT_TLV_* defines. When setting Type=interface status is desired, the flag BCM_OAM_ENDPOINT_INTERFACE_STATE_UPDATE should be set in the flags field and the interface_field should be set to one of the BCM_OAM_INTERFACE_TLV_* defines. The OAMP may also monitor the port/interface status of received CCMs. This may be done via the bcm_oam_endpoint_get() API for the remote MEP entry. The value of the latest port status will be returned at the field port_state and the value of the latest interface status will be returned at the field interface_status. Likewise for each port/interface value an interrupt may be triggered upon reception of the first CCM with such a value. This can be achieved using the API bcm_oam_event_register(). Note that due to a limitation at the OAMP a program at the egress editor must be used for CCMs destined to the OAMP. The program selection is based on an egress PMF rule. An example of such a rule may be found in cint_oam_field_tlv.c For a more detailed explenation consult the user manual.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60657	818481	88660_A0 88670_A0	OAM: Support OAMP server. For complete documentation consult the user manual, or for an example see oamp_server_example() in cint_oam_acceleration_advanced_features.c. Currently only CCM support is available. Note that down MEP CCM transmission may also be defined on LAG gports.
SDK-60670		56850_A0 56850_A1 56850_A2	Support has been removed for API to retrieve member port for DLB of HG Trunk.
SDK-60672	816228	88650_A0	Info log should not flood the console with log message. Now change the log severity from verbose to debug and info to verbose can fix this issue.
SDK-60692	819679	56640_A0 56540_A0	Problem : This is due to the issue that when we detach a counter from field entry, field module will reduce the field_specific reference count but it was not updating the stat module specific reference count . Due to this stat destroy was throwing error saying reference count is still not zero. Solution : Field entry detach routine of counter was updated to call relevant function to reduce stat related reference count.
SDK-60710	794588	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In the previous release, the API BCM_PORT_EXTENDER_TYPE_NONE was not implemented. In this release, this issue has been addressed by implementing the API.
SDK-60711	817525	All	Handle LinkScan in case of port removal and insertion. Not relevant for most customers
SDK-60719 SDK-52048	818562	56640_A0	Fixed following issues related to OAM warmboot recovery: - recovery tries to access an invalid field - memory pointer getting freed was not set to NULL. - CCM period was not being recovered properly - ma index re-allocation was being attempted while trying to recover endpoint on same vlan port but different level, which resulted in failure. - scache allocation was not correct during upgrade from older version.
SDK-60729	818782	88660_A0 88670_A0	The bcmFieldQualifyVlanFormat is exposed at Egress Field Processor stage for both key and preselection. Its encoding is different, according to the internal Ethernet-Tag-Format (5 bits) signal.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60750		88660_A0	In L2 MAC table, aging period was not configurable per VSI. The number of meta-cycles of an entry not being observed before it is deleted were hard-coded. The number of aging meta-cycles before entry aging can now be configured per VSI, by using the API function bcm_vlan_control_vlan_set() and setting the field aging_cycles in the input structure bcm_vlan_control_vlan_t. Valid range for aging_cycles: 0-6, 0 is default. The user may configure up to 4 different aging cycle values for different VSI's.
SDK-60756	816538	56340_A0	Loopback ports were not being to valid ports bitmap for helix_4 in previous releases, and hence any operations on loopback ports were failing. This is now fixed by adding the loopback ports to valid ports bitmap.
SDK-60769	816767	56850_A0 56850_A1 56850_A2	In the previous release, the TPID did not enable VLAN GPORT. In this release, this issue has been addressed by correcting the related APIs.
SDK-60784	818772	56334_A0	In the previous SDK, the oam cli command could not accept MiM gport for port parameter when creating endpoints. This has been changed to allow to create endpoints on MiM gport.
SDK-60797	820536	56850_A0 56850_A1 56850_A2	Fixed bug in interrupt handler for CMICm-based devices which would inadvertently enable hardware interrupts in polled-IRQ mode thus causing a system hang.
SDK-60843		56450_A0 56340_A0	Enabled internal termination on the gigabit serdes interface for HX4 & KT2 when system is operating in eHost mode.
SDK-60846	809695	56440_A0 56440_B0	For BCM 5644x devices there was an issue with parity errors coming from SVM_POLICY_TABLE and SVM_MACROFLOW_INDEX_TABLE entry. This has been fixed.
SDK-60853	819575	56540_A0 56540_B0	True Egress frames of size greater than 8K were not getting egressed. Reason for this was that MTU Size programmed in PASSTHRU_NLF_MTU_CHECK register was set to 8K leading to drop seen. This default MTU Size on loopback port has been increased to maximum value which is around 16K by default . This is done to be in sync with other MTU register which currently hold default value of 16K.
SDK-60860		56844_A0	Add support for SKU 56742
SDK-60890		56845_B0 56845_A2 56840_A0 56850_A0 56850_A1 56850_A2	In previous release, bcm_qos_port_map_set() did not set TRUST_DSCP_V4/V6 on physical ports but set it on virtual port. The different behavior in the same API made customer confused, now this issue has been fixed.
SDK-60900	820836	88650_B1 88660_A0	In Forwarding stage (FLP), the Mac-in-Mac and TRILL-Multicast FLP programs were allocated statically at init with the same program ID. This prevented to allow both applications to run simultaneously. The Mac-in-Mac FLP program is now allocated dynamically.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60905	821319	56450_A0 56450_B0	In the previous release the validity check for the nodes in LLS tree was missing, this caused a segmentation error if any invalid node was getting accessed while flushing it. The check has been added in this release to avoid any segmentation error in attempt of accessing invalid nodes in the LLS tree.
SDK-60909		88650_A0	Minor coverity issues resolution involving the removal of unreachable code, and uninitialized variables.
SDK-60927	819752	88030_B0	BCM_CALADAN3_SUPPORT was being used without SOC_IS_CALADAN3(unit) check thus affect ARAD or other images being built with the above #ifdefine.
SDK-60955		All	Configuration variables which are parsed before soc_attach() is called, no longer require that the unit number is appended. For example, polled_irq_mode=1 now works as expected, whereas before this change it had to be specified as polled_irq_mode.<unit>=1.
SDK-60956		88650_A0	Diag command "diag cosq" without any parameter crashes. The command fixed to return usage description.
SDK-60958		88650_B1 88660_A0	Corrective action for IPS QDESC parity error: A parity error in the IPS QDESC dynamic memory, corresponding to an unused VoQ, while credit watchdog range includes this unused VoQ, would cause a parity interrupt that is never fixed. Eventually the interrupt handler would detect interrupt storm and mask IPS parity interrupts. This fix recovers for a SER on unused VoQ by writing (zero-value) to the corrupted entry. Please note that for entries corrsponding to active VoQ, SER condition is resolved once the ASIC updates the appropriate entry (no change from current behavior for active VoQs)
SDK-60964	820443	56640_A0 56340_A0	In previous releases, when parity errors are detected in VLAN_XLATE_1 or VLAN_XLATE_2 tables in Triumph_3 or Helix_4, the table name was not resolved in SER correction routine. This would cause VLAN_XLATE_1 or VLAN_XLATE_2 tables not to be corrected correctly. In this release, this bug has been fixed. Table name will be returned correctly and table entry will be corrected once parity error is detected.
SDK-60971		56845_B0 56845_A2	In earlier release, the base_ptr field in ECMP group table was not configured correctly on TD+ during defragment. This has been resolved.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-60988	821704	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In earlier releases, deleting a secondary L3 Interface unbinded the VRF from the primary one. This has been resolved.
SDK-61031	823863	All	Host endian-ness was not correctly handled in SDK side as-well-as in Ukernel side. Due to that where byte swapping was not required (on Little Endian eHost) but was happening causing all reverse data,string messages. Ukernel Side fix: Corrected endianess issue and added some print messages for debugging purpose. Now Bytes will be swapped only when host side(ehost) endian-ness and own(ukernel) endian-ness differs. SDK Side fix: 1)Removed version string display in init phase as endian-ness of host side is not known in init-phase i.e. after system-info reply only, ukernel recognize and sends correct version-info string 2)Removed hack of reverse version-string on LE host 3)Now sends 0 for LE host and 1 for BE host as system info reply
SDK-61032	815490	56440_A0 56440_A1 56450_A0 56440_B0 56450_B0	In Katana & Katana2 devices, DDR Phytune command was causing high CPU utilization and parity errors during SDK init. This was because of parity errors in Channel interfaces were not handled and were also checked for interfaces that were not used. Fixed this issue by modifying CI error mask to handle parity errors and also enabled partiy checks for only those interfaces supported in device
SDK-61063		56340_A0	There was memory leak inside `bcm_regex_match_set()` API in earlier release which was causing all system memory to be consumed when `bcm_regex_match_set()` was called repeatedly for AppIQ regex signature configuration,Fixed this API to avoid memory leak by freeing allocated memory when not in use.
SDK-61073	813684	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In the previous release, `BCM_L3_TRILL_ONLY`/ `BCM_L3_VXLAN_ONLY`/ `BCM_L3_L2GRE_ONLY` were not saved. In this release, these flags are saved now.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-61093		88650_A0 88650_B0 88650_B1 88660_A0	Fix stat_if_pkt_size documentation in config files.
SDK-61094	822899	56640_B0	Description: EGR_VLAN_XLATE table can't be recovered when the flow continue hit the entry Root cause: When the flow hits a corruted EGR_VLAN_XLATE entry continously, the EGR_VLAN_XLATE SER ERROR interrupt should be raised and then SER handler would be invoked to correct it. Unfortunately, there is another SER ERROR interrupt being raised meanwhile which notified SOP CELL error and packet drop happened. Every corrupted EGR_VLAN_XLATE entry hit will trigger this interrupt until the EGR_VLAN_XLATE ser error has been corrected. So the interrupt handler may go into a dead loop of handling the continously SOP CELL error and packet drop interrupt since there is no opportunity to correct the corrupted EGR_VLAN_XLATE entry. Solution: Current software has changed the code to ensure handling EGR_VLAN_XLATE entry correction before the SOP CELL error notification.
SDK-61096		56340_A0	In earlier releases, there was no memory limit on the allocation for regex expansion which causes memory corruption in othe applications as well. Boundary has now been set for not more than 100MB.
SDK-61144		56063 56064 56062 56060 53400_A0 53406_A0 53402_A0	Apply the software workaround as suggested for 10G TSCPLL for the case that strap status of lcpll1_refclk_sel equals to 0 on Greyhound platofrms.
SDK-61171	823889	56340_A0	Issue :- bcm_ipmc_add() calling failure after SDK upgrade to 6.3.7 from 6.3.4. Fix :- Issue was due to Code changes that were added to verify if entry is not present and REPLACE flag is passed, the API should return BCM_E_NOT_FOUND. Reverted the code change.
SDK-61199	819962	All 56850_A0 56850_A1 56850_A2	Added support for transmitting ETAG packets out of the extender port without modifying the ETAG.
SDK-61244	824828	88660_A0 88670_A0	In Field processor, when creating a new entry for direct extraction Database, the entry ID is taken from a range of Direct extraction pull, which comes after the TCAM ID pull. Thus, this range is offset by the number of TCAM entries (Internal and external). This maximum number of TCAM entries was defined incorrectly. It is fixed.
SDK-61251	814934	88660_A0	If there are 2 task to call bcm_l2_learn_limit_set(), the limit number in ARM wasn't same with the number at SDK at sometimes. Now add a lock to fix the issue.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-61261		88660_A0 88670_A0	BFD: Adding BFD echo support. To configure BFD echo session the following sequence should be used: 1) Enable BFD echo soc property bfd_echo_enabled (together with all other oam soc properties) 2) Call bcm_bfd_init right after the device was loaded 3) Register callbacks for Loss-Of-Continuity. 4) Configure ip session as described in user manual 5) Add IP host with destination bcmRxTrapBfdEchoOverIpv4 6) Add PMF rule as shown in cint_bfd.c example, function bfd_echo_field_trap_qualifier 7) Add bfd control endpoint as described in user manual 8) Add bfd echo endpoint by calling bcm_bfd_endpoint_create with the same fields as in number 7 and set the flags BCM_BFD_ECHO & BCM_BFD_ENDPOINT_MULTIHOP.
SDK-61267		88650_B0 88660_A0	In FCoE VSAN assignment, when mode was updated to "VSAN from VSI" before enabling the NPV mode the VSAN assignment is not updated correctly. This is fixed.
SDK-61270		88650_B0 88650_B1 88660_A0	In FCoE, VSAN mode update did not updated the NPV programs in the FLP. Now, it is updated when NPV is enabled.
SDK-61276		88660_A0	OAM: In the APIs bcm_oam_loss/delay_add(), Fixed a bug concerning the field peer_da_mac_address. Under certain circumstances MSBs of the DA address of outgoing LMMs/DMMs wasnt properly set.
SDK-61281	811786	56456_B0 56640_A0 56450_A0 56640_A1 56640_B0 56455_A0 56456_A0 56450_B0	Fix contains changes for bcm_port_timesync_config_get() API which has a minor variable intialization error causing breaking of API functionality. Also syntax for deleting timesync configuration using bcm_port_timesync_config_set() API is documented.
SDK-61282		All	Diagnostic shell now prints timestamp with PTP events.
SDK-61301	813742	88650_A0 88660_A0	In LAG and ECMP, Load Balancing is not working according to native Ethernet for VxLAN packets. This is fixed.
SDK-61316		88650_A0 88650_B0 88660_A0	In FCoE application, key construction for LPM lookups in forwarding stage (FLP) when working with VSAN from VSI mode was built wrong. key construction fixed to be consistent with entries added to DB.
SDK-61317		88650_A0 88650_B0 88660_A0	OAM: When using bcm_oam_action_set, deletion and creation of new endpoint was failing after a few iterations.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-61323		88650_A0	BFD: For BFD endpoints of type other than bcmBFDTunnelTypeUdp, bcmBFDTunnelTypeMpls, if the flag BCM_BFD_ENDPOINT_IN_HW is not set (endpoint is not accelerated to the OAMP) then the field local_discr is not used. Thus this field must be zero when bcm_bfd_endpoint_create() is called in such cases. Likewise in all cases where the field local_discr is required, upon calling bcm_bfd_endpoint_create() correctly verify that the MSBs are consistent with prior endpoints added (The MSBs act as a global value per device). For further detail consult the user manual.
SDK-61373		88650_A0	BFD: Enable configuration of different BFD types (IP/MPS/MPLS-TP) on same device.
SDK-61381	825267	All	RX LOS application is used to detect, monitor, and manage RX loss of signal (LoS) indications from the BCM88650 and BCM88750 SerDes interfaces. Function rx_los_unit_detach used to detach unit from application and mask all the relevant to interrupts. Added a function (rx_los_sw_unit_detach) which allows to perform SW detach without any access to device (skip interrupts masking)
SDK-61406 SDK-61530		56440_A0	A scheduling change failed after warmboot if dynamic sched change algorithm was enabled.The missing warmboot data is now constructed and able to change sched mode after warmboot .
SDK-61442		88650_A0 88650_B0 88650_B1 88660_A0	Counter thread need to count for all ILKN channels when counting in per channel mode. Fixed.
SDK-61477	810833	88650_B1 88660_A0	As part of SER interrupt handling, the SDK can attempt to write to the corrupted memory. This is not allowed for dynamic memories, i.e. memories that are maintained by the device and not by CPU configuration. The interrupt handling logic did not account for this limitation. For most memories, trying to override a dynamic memory is protected by HW and would fail with error message. Some dynamic memories are not protected, writing to these memories could result in unpredictable device state. The fix does allow caching of 2 static memories which listed at dynamic list by mistake: OAMP_RMEP_DB OLP_DSP_EVENT_ROUTE
SDK-61484		56514_A0	During fp re-installation removed action items from software were not getting deleted in hardware.Reason for this behavior was during re-install action items marked as invalid were not skipped . Added a check to skip invalid action items from re-installation.
SDK-61486	826612	88650_B1 88660_A0	cint_vswitch_vpls_roo.c is fixed to configure correct Link Layer (instead of Dst MAC 0)

Table 100:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.2*
SDK-61503		88650_A0	OAM: 88650_A0 does not support 1588 timestamp format. Validity check is added in bcm_oam_endpoint_create().
SDK-61505	826710	56334_B0 56334_A0	Add bcm_switch_pkt_info_hash_get API support for Enduro.
SDK-61507		88660_A0	BFD: For BFD over IP packets generated by the OAMP the UDP checksum should be set to 0 (none) as opposed to 0xffff.
SDK-61526		88750_A0 88750_B0	BCM88750 standalone compilation might fail due to linkage errors. Fixed.
SDK-61527		88660_A0	MPLS PORT: In case of ingress only configuration, learning was not configured properly.
SDK-61545	827380	56440_B0 56240_B0	In the earlier release, the "vlan translate action add" was not working when "policer" was not specified. This was because the default value of Policer was being set to "None" which is not its default value. The issue has been fixed by setting the default value to 0 in the current release.
SDK-61613		88660_A0	OAM InLIF profile can now be mapped to 4 default OAM trap profile. Each such trap profile can define a default endpoint to trap OAMoEth packets arriving on an InLIF with the suitable profile. Call sequence: 1. bcm_port_control_set - to map an InLIF profile to a OAM-Trap-profile (In simple InLIF profile mode, this sets the InLIF profile bits) 2. bcm_oam_endpoint_create with: - id = BCM_OAM_ENDPOINT_DEFAULT_INGRESS 0/1/2/3 (depending on the OAM-Trap-profile chosen in the previous step) - level = packets with MDL<=level will be trapped by that endpoint A default Up-MEP can also be created using: 1. bcm_oam_endpoint_create with: - id = BCM_OAM_ENDPOINT_DEFAULT_EGRESS0 - level = packets with MDL<=level will be trapped by that endpoint - flags |= BCM_OAM_ENDPOINT_UP_FACING for example sequence see cint_oam.c
SDK-61616		88670_A0	In BCM88670, QOS support for ROO application. Add support for PCP DEI profile in ROO application, using a new PCP DEI profile table. PCP DEI profile table: for a PCP DEI profile get the indication if the PCP DEI is mapped from DSCP or mapped from TC DP.
SDK-61619		88650_A0 88660_A0 88650_B0	For external Tcam (KBP) connection, use this soc property (ext_ilkn_reverse) to indicate the numbering mode of the lanes in KBP. default value for the system is 1 (reversed).
SDK-61635	828430	88030_A0 88030_B0	Note.
SDK-61636	828428	88030_A0 54880_B0	Fixed
SDK-61641	827595	56850_A0 56850_A1 56850_A2	In earlier releases, port control bcmPortControlSerdesReset was effective only once, after a power-on-reset. That's because the code did not reset TSC, only took it out of reset. This has been fixed in this release by resetting TSC first then taking it out of reset.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-61652	826180	56640_A0 56640_A1 56640_B0	Problem Statement: The switch was hanging during exit clean with multiple VLAN_XLATE entries. Resolution: The hang was due to an uninitialized variable that was causing infinite loop. The variable is now initialized so that it picks appropriate value.
SDK-61659	828631	56450_A0 56450_B0	In the earlier release, a memory corruption issue was seen due to lesser memory allocation which did not accommodate the PP ports. The issue has been fixed by increasing the allocation size to include the PP ports in the current release.
SDK-61663		88650_A0	OAM: For Y.1731 over MPLS, allow classification of DMMs/DMRs with the field Version on the CFM header set to 1 (this is the value defined by Y.1731).
SDK-61672		88670_A0	MPLS: In case OAM is not enabled on the device, Router Alert is not set as special label. It can be trapped using the lif table as a non-special label.
SDK-61727	796172	All 56850_A0 56850_A1 56850_A2	In the previous release, when updating the extender port id which had been added into a VPLAG, the field SOURCE_VP in the table VLAN_XLATE would be mistakenly modified from the VP of the VPLAG to the VP of the extender port id, which was incorrect. In this release, this issue has been addressed by updating the field SOURCE_VP in the table VLAN_XLATE with the VP of the VPLAG.
SDK-61813	825520	56440_A0 56440_B0	In the earlier version, entry pointed by EGR_VLAN_CONTROL_1r and TPID reference count was not handled correctly. The fix correctly handles the problem and the issue is not seen.
SDK-61818	830940	56846_A0 56640_A0 56850_A0	In the previous release, the API bcm_multicast_egress_set couldn't delete the given L3 INTF from a multicast group with a specific sequence. In this release, this issue has been addressed by correcting the internal function of comparing the software status and hardware resources.
SDK-61823		88650_A0	Upon SER event, the driver collects information identifying the affected table entry. This information was not sampled correctly for IQM Parity-protected tables, and for OLP ECC-protected tables. As a result, prior to this fix parity errors in any IQM table, as well as ECC errors in any OLP table, were not handled by the appropriate corrective-action. However, the SER event detection and reporting functioned properly.
SDK-61855	831077	56834_A2	In previous releases, the L2_entry table size was fixed to 160K. The modifcation was to allow user to set the L2_entry table size to the capacity under 160K.
SDK-61869		88660_A0 88670_A0	When creating OAM default endpoint in the egress, use the ID defined as BCM_OAM_ENDPOINT_DEFAULT_EGRESS0 for BCM88660 BCM_OAM_ENDPOINT_DEFAULT_EGRESS0/1/2/3 for BCM88670 and above.

Table 100:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.2*
SDK-61883		`88660_A0`	OAM: New oam classifier mode is available. It enables adding up to 8 MEPs and up to 2 MIPs on each lif, in any direction (up/down), but reduces available amount of endpoints to 8K. To enable the new mode set soc property `oam_classifier_advanced_mode` to value 2.
SDK-61901		`88650_B1`	Added "init `rx_los"` to init commands usage.
SDK-61915		`88660_A0`	OAM: Incorrect mp-profile allocation at the OAM classifier may cause incorrect behavior of endpoints set on different directions of the same lif.
SDK-61934		`88660_A0`	OAM: Fixed bug causing `bcm_oam_loss_add/delete(),` `bcm_oam_delay_add/delete()` to misbehave. Error may be returned after any several calls to add APIs, each for different endpoints.
SDK-61938		`56640_A0 56640_A1 56640_B0`	Internal phy driver was not able to attach to few XTPORT block ports of triumph3 devices. A change made for a separate issue was blocking PLL reset for few block of the ports. This has been resolved
SDK-61949	830803	`56850_A0 56850_A1 56850_A2`	Previously, due to sharing of the fundamental interrupter handler, bcmINTR would be woken up on each received knet interrupter even though there wasn't any one interrupter that needed to be processed by bcmINTR. This reduced the system performance. This has been resolved by identifying the interrupter's owner, the fundamental interrupter handler won't wake up bcmINTR thread if there isn't bcmINTR-related interrupter.
SDK-62000	815947	All	Fixed usage of allocated-but-not-initialized memory by PTP module.
SDK-62002		All	Fixed enabling using the RPT link-integrity bitmap with IreTdmMask, only when calling `bcm_fabric_tdm_direct_routing_set(),` and not on init. In register `FDT_LINK_BITMAP_CONFIGURATIONr` field `IRE_TDM_MASK_MODEf,` The default mode is 2 - which means using RTP reachable bitmap. When calling `bcm_fabric_tdm_direct_routing_set()` The mode changes to 3 - which means using RTP link-integrity bitmap.
SDK-62018	826377	`88650_B1 88660_A0`	A bug caused extra bytes to be added to RSPAN packets in case a user header was used.
SDK-62022		`88660_A0`	MPLS PORT: Push-Profile allocations are mismanaged in PWE ingress only mode.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-62025 SDK-59407	832310	56850_A0 56850_A1 56850_A2	In earlier releases, when multiple trunk member ports were in failed state, deleting trunk member threw error. This has been fixed in this release by just skipping hardware configurations and returning ok, when multiple trunk member ports are in failed state. Fix a bug that when multiple trunk member ports are in failed state, deleting trunk member will throw error.
SDK-62032		88650_A0 88660_A0 88670_A0	In LAG and ECMP, Load Balancing is not working according to native IP for VxLAN packets. This is fixed.
SDK-62062	829166	88650_A0 88650_B0 88650_B1 88650ACP_A0 88660_A0 All	MPLS PORT: Flag BCM_MPLS_EGRESS_LABEL_TTL_DECREMENT is no longer mandatory for bcm_mpls_port_add api with action PHP.
SDK-62063		88650_A0 88650_B0 88650_B1 88660_A0	Port protocol based vlan classification. Configuration of vlan per port and trill/mpls ethertype, using the api bcm_vlan_port_protocol_action_add was causing a new user defined ether_type protocol, instead of being identified as trill or mpls ethertypes.
SDK-62102	832410	53284_A0	Fixed the issue on ROBO chips thin which vlan information was not restored to default STG when removing the vlan from other STG.
SDK-62106	832908	56450_A0 56450_B0	In earlier releases INNER_TPID_ENABLE field in SYSTEM_CONFIG_TABLE was not configured during initialization for Katana2. This issue has been resolved now by setting INNER_TPID_ENABLE to 1 during port initialization.
SDK-62124		88750_A0 88950_a0	FE have limitations on its local modid values: BCM88750: 0-127 BCM88950: 0-143 But the limitations for FE13 local modid values are different, so FE13 the limitations have been changed to: BCM88750: 0-63 BCM88950: 0-71
SDK-62126		88650_B1 88660_A0	In L3, when using internal LPM routing table, adding and updating routes may cause memory overruns: Fixed possible memory overrun in DMA buffers or in adjacent memory. The memory overrun was up to 240 bytes after the buffer for route adds.
SDK-62209	834157	88650_B1 88660_A0	L3: L3 egress can be created by calling bcm_l3_egress_create. When creating L3 egress with voq_mapping_mode=DIRECT and egress.port (system-port) is bigger than 4K, API return ARAD_PP_FRWRD_FEC_DEST_VAL_OUT_OF_RANGE_ERR. L3 egress can be created correctly after the fix.
SDK-62222	826150	All	In the previous release, the BCM SHELL command "mc show" showed some interfaces in the given group which they didn't belong to, which was incorrect. In this release, this issue has been addressed by correcting the internal function of comparing the software status and hardware resources.

Table 100:

Number	CSP #	Chips	Release Notes For 6.4.2
SDK-62227		88650_A0	BFD: For incoming packets classified as BFD IP-multi-hop (according to the UDP dest-port), packet's destination IP address is compared to the endpoint's source IP address, which is configured in the field src_ip_addr in the API bcm_bfd_endpoint_create(). Up to 16 different source IP addresses are available per device.
SDK-62296		56440_A0 56850_A0	PTP subsystem now supports stack deletion, firmware reload, and restart.
SDK-62333		88660_A0	BFD: InLIF profile can now be mapped to 4 default OAM trap profile. Each such trap profile can define a default endpoint to trap BFD packets arriving on an InLIF with the suitable profile. Call sequence: 1. bcm_port_control_set - to map an InLIF profile to a OAM-Trap-profile (In simple InLIF profile mode, this sets the InLIF profile bits) 2. bcm_bfd_endpoint_create with: - id = BCM_BFD_ENDPOINT_DEFAULT0/1/2/3 (depending on the OAM-Trap-profile chosen in the previous step) - remote_gport = port to trap to (defaults to the CPU port) for example sequence see cint_bfd.c
SDK-62336		88660_A0	OAM: When a MEP is configured on a LIF after a MIP is already configured on it, device might not handle MIP packets as expected.
SDK-62355		88650_A0 88650_B0 88650_B1 88650ACP_A0 88660_A0 88670_A0	MPLS PORT: Enable replacing encap_id in case PWE is protected
SDK-62364		88650_A0 88660_A0 88670_A0	Direct ingress queues to rate class mapping, and using an internal mechanism to handle the mapping is mutually exclusive. Blocked using direct ingress queues to rate class mapping after using the internally handled mechanism is blocked. relevant API: bcm_cosq_profile_mapping_set(int unit, bcm_gport_t gport_to_map, bcm_cos_queue_t cosq, uint32 flags, bcm_switch_profile_mapping_t *profile_mapping); MACROs: BCM_GPORT_PROFILE_GET/SET, BCM_GPORT_IS_PROFILE;
SDK-62367		88650_A0	OAM: Added "diag oam exact_match LIF=N" : display O-EM 1 entry by LIF
SDK-62421		88660_A0	OAM: Inconsistent behavior might be seen on LIFs on which MIPs and MEPs were created and action set was used to modify destination.
SDK-62432		88660_A0	OAM: Calls to bcm_oam_loss/delay_delete() may interfere with the DA address of other available LMM/DMMs transmitted by the OAMP.
SDK-62437	836260	88750_A0	BCM88750 load failed when soc property backplane_serdes_encoding is not explicitly defined. Fixed. In such a case BCM88750 encoding will be set to "KR_FEC",

Table 100:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.2*
SDK-62492		88660_A0	OAM: when creating accelerated endpoints of type bcmOAMEndpointTypeBHHMPLS, the MEP DB entry may not be properly updated, causing subsequent calls to bcm_oam_loss/delay_add() to fail.
SDK-62533	835794	56450_B1 56450_A0 56450_B0	Issue: Field InPorts Qualifier for max supported ports on KT2 device was failing to recover during level-1 Warmboot. Root Cause: During recovery, the group is constructed for all the ports in the device by retrieving from a internal macro which is updated by a port structure. However, during init time, the flex ports were not part of this macro due to which, the group pbmp and hardware configured pbmp value mismatches and returns an internal error. Fix: During recovery, the group is constructed for all the supported ports in the device by retrieving the flex ports explicitly.
SDK-62548		88660_A0	OAM: When a MIP is deleted, if no other MIP is defined on the same LIF, the remaining MEPs might not act as expected.
SDK-62721		88650_A0 88660_A0	OAM: calls to bcm_oam_endpoint_create with the flags BCM_OAM_ENDPOINT_REPLACE and BCM_OAM_ENDPOINT_REMOTE may fail due to uninitialized values.
SDK-62817		88650_A0 88660_A0	OAM: Remove duplication, unused code from cint_oam.c
SDK-62820		88660_A0	OAM Classifier behavior is inconsistent after adding and removing endpoints on same lif.

Section 6: Unresolved Issues for 6.4.2

The following issues are unresolved in version 6.4.2 of the SDK.

Table 101:

Number	***CSP #***	***Chips***	***Release Notes***
SDK-30856		All	When mirror-to port resides on a different unit, the mirrored packet may not egress correctly.
SDK-33686	389108	`56634_A0`	If a MiM virtual port has statistics enabled for it and if such MiM port is replaced using BCM API `bcm_mim_port_add( )` along with flag `BCM_MIM_PORT_REPLACE` then the statistics of that MiM port might be lost.
SDK-35755	411572	`56820_A0 56820_B0`	Compared to older releases, L2 Notification thread (bcmL2X) requires more CPU bandwidth to run in polling mode `(l2xmsg_mode=0),` due to the requirement for more thorough entry comparisons. The recommendation, however, is to run L2 notification thread using `L2MOD_FIFO` DMA mechanism, which is much more efficient and provides more functionality. To do that, please, set the configuration variable (property) `l2xmsg_mode` to 1.

SDK-37821		`56846_A0 56845_B0`
		`56845_A2 56844_A0`
		`56842_A0 56840_A0`
		`56440_A0 56843_B0`
		`56841_A3 56846_A1`
		`56841_B0`
			`bcm_cosq_config_set()` had traditionally been used to set the system wide number of COSQs. This does not apply to devices with hierarchical schedulers. For these devices, the hierarchy constructed at device initialization time is dependent upon the number of COSQs defined in the system configuration at the time of initialization. Changing the queue count after the hierarchy has been constructed has no effect.

SDK-44416		`88640_A0`	1. API is reading the wrong register from the device. 2. API is missing the parameter of ResetLoad, so this value cannot be configured.
SDK-44471	599747	`56544_A0`	BCM56544 XAUI ports support single lane GE operation via lane 0 (at boot time). The applicable config is `bcm56544_4x10_12x10=1.` However, current software has not supported this yet. Modifying the `src/soc/esw/triumph3.c->port_speed_max_94` as following can support GE operation: static const `int port_speed_max_94[]` = { -1, 1/* 10 */, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1/* 10 */, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1/* 10 */, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1/* 10 */, -1, -1, -1, -1, -1, -1, -1, 10, 10, 10, 10, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 10, 10, 10, 10, 10, 10, 10, 10, -1, -1, -1, -1, -1 }; However, there should be more decent way to achieve this feature.
SDK-45075		All	When an interrupt occurs on different blocks of the same type (e.g. multiple FMAC blocks), the count will be accumulated in the same counter. For example RX-LOS interrupt may occur on different FMAC blocks, and counted as same recurring event, although it is in fact a different interrupt. This may affect corrective action in case it is different for a recurring event, in the case if recurring-threshold for this event is crossed.

Table 101:

Number	CSP #	Chips	Release Notes
SDK-45366	611273	56440_A0	When the API bcm_cosq_port_bandwidth_set(...) is called on a particular port and COSq to enable egress rate limiting, sometimes the CLI command "show c" will show the incorrect dropping statistics on a irrelevant port.
SDK-46556	621213	88650_A0 88650_B0 88650_B1	bcm_l2_cache_delete() API does not delete general_trap entry configuration in HW
SDK-47366	642398	All	The implementation of SER (Soft Error Recovery mechanism) requires the SDK to perform periodical scanning of certain memories. The infrastructure for this scanning is provided by the optional MEM_SCAN feature (component) of the SDK. Since SER is a mandatory component, that can't be compiled out, MEM_SCAN becomes a mandatory component too as long as you are using a device, supported by SER.
SDK-47739	628786	56540_A0 56540_B0	For devices in the BCM56540 family, the CPU queues are allocated differently depending on the revision of the device (Ax vs. Bx). This force the developer to include revision specific code in the application.
SDK-48091	662661	56850_A0 56850_A1 56850_A2	For BCM56850 devices, when only a single GigE port is allocated to a TSC lane (the other 3 TSC lanes are not used), that port may be configured incorrectly resulting in port that appears functional but is not.
SDK-48913		88650_A0 88650_B0 88650_B1	In RX Trap API, setting a new Counter-Pointer value may not take effect. Under investigation.
SDK-51978		88650_A0	In a device with channalized CPU ports, where some of the CPU ports are Higig and some not, the WB wont preserve the Higig indication correctly.
SDK-52383		88650_A0 88650_B0	Cud extension for Arad is not supported
SDK-54219	744517	88650_B1	In ARAD B1, VxLAN/L2GRE packet size of 236B-299B will be dropped by EPNI if the packet needs to be terminated. Workaround is introduced in FP CINT: cint_field_ingress_large_termination.c
SDK-54488		88650_A0	In Field Processor, when creating a cascaded field group, a failure might occur in key allocation in case the key ID is already determined. To be investigated.
SDK-54623		88660_A0	In Field Processor, when creating a field group, false error messages are printed when operation succeeds (return value indicates success). To be fixed.

Section 7: Device and Platform Support

The section describes all devices, platforms, and operating systems that are supported by this release.

SWITCH DEVICES

Table 102: Switch Devices

Family	Devices	Description
BCM5389	BCM5389 A0	8-Port GbE Switch with Integrated Serdes
	BCM5389 A1	8-Port GbE Switch with Integrated Serdes
BCM5396	BCM5396 A0	16-Port GbE Switch with Integrated Serdes
BCM53010	BCM53010 A0	5-Port Gigabit Ethernet Managed Switch integrated with single core ARM Cortex-A9 processor
	BCM53010 A2	
	BCM53011 A0	5-Port Gigabit Ethernet Managed Switch integrated with dual cores ARM Cortex-A9 processor
	BCM53011 A2	
	BCM53012 A0	5-Port Gigabit Ethernet Managed Switch with one RGMII I/F integrated with dual cores ARM Cortex-A9 processor
	BCM53012 A2	
BCM53018	BCM53017 A0	2-Port Gigabit Ethernet Managed Switch with one RGMII I/F integrated with dual cores ARM Cortex-A9 processor
	BCM53018 A0	5-Port Gigabit Ethernet Managed Switch with one RGMII I/F integrated with dual cores ARM Cortex-A9 processor
	BCM53019 A0	5-Port Gigabit Ethernet Managed Switch integrated with dual cores ARM Cortex-A9 processor
BCM53020	BCM58522 A0	5-Port Gigabit Ethernet Managed Switch integrated with 2 PHYs, ARM Cortex-A9 processor and macsec cores
	BCM58525 A0	5-Port Gigabit Ethernet Managed Switch integrated with 2 PHYs, SGMII I/F, ARM Cortex-A9 processor and macsec cores
	BCM58622 A0	8-Port Gigabit Ethernet Managed Switch integrated with 5 PHYs, ARM Cortex-A9 processor and macsec cores
	BCM58623 A0	8-Port Gigabit Ethernet Managed Switch integrated with 5 PHYs, ARM Cortex-A9 processor and macsec cores
	BCM58625 A0	8-Port Gigabit Ethernet Managed Switch integrated with 5 PHYs, SGMII I/F, ARM Cortex-A9 processor and macsec cores
BCM53101	BCM53101 A0	5-Port Fast Ethernet Managed Switch + 1 Fast Ethernet WAN port
	BCM53101 B0	
BCM53115	BCM53115 A0	5-Port GbE Managed Switch + 1 Gigabit WAN port with integrated serdes
	BCM53115 A1	
	BCM53115 B0	
	BCM53115 B1	
	BCM53115 C0	
BCM53118	BCM53118 A0	8-Port Gigabit Ethernet Switch
	BCM53118 B0	
	BCM53118 B1	
BCM53125	BCM53125 A0	5-Port Gigabit Ethernet Switch with 1 Gigabit WAN port and 8051 processor
	BCM53125 B0	
BCM53128	BCM53128 A0	8-Port Gigabit Ethernet Switch with embedded 8051 processor
	BCM53128 B0	
BCM53242	BCM53242 A0	Managed Switch with 24 FE Ports + 2 GbE Interface
	BCM53242 B0	

Table 102: Switch Devices

Family	Devices	Description
	BCM53242 B1	
	BCM53262 A0	Managed Switch with 24 FE Ports + 4 GbE Interface
	BCM53262 B0	
	BCM53262 B1	
BCM53280	BCM53282 A0	8-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch
	BCM53282 B0	
	BCM53282 B1	
	BCM53282 B2	
	BCM53283 A0	16-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch
	BCM53283 B0	
	BCM53283 B1	
	BCM53283 B2	
	BCM53284 A0	24-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch
	BCM53284 B0	
	BCM53284 B1	
	BCM53284 B2	
	BCM53286 A0	24-Port Fast Ethernet + 4-Port Gigabit Ethernet Multilayer Switch
	BCM53286 B0	
	BCM53286 B1	
	BCM53286 B2	
	BCM53288 A0	24-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch with one 2.5GbE Uplink Port
	BCM53288 B0	
	BCM53288 B1	
	BCM53288 B2	
BCM53300	BCM53300 A0	Managed 24-port L2 Switch
	BCM53300 A1	
	BCM53301 A0	Managed 16-port L2 Switch
	BCM53301 A1	
	BCM53302 A0	Managed 24-port L2 Switch
	BCM53302 A1	
BCM53310	BCM53312 A0	BCM53312 Integrated Multilayer Switch and CPU
	BCM53312 B0	
	BCM53313 A0	BCM53313 Integrated Multilayer Switch and CPU
	BCM53313 B0	
	BCM53314 A0	BCM53314 Integrated Multilayer Switch and CPU
	BCM53314 B0	
BCM53320	BCM53322 A0	BCM53322 Integrated Multilayer Switch and CPU
	BCM53323 A0	BCM53323 Integrated Multilayer Switch and CPU
	BCM53324 A0	BCM53324 Integrated Multilayer Switch and CPU
BCM53400	BCM53406 A0	12-port 10GbE plus 8-port 2.5GbE and 4-port 5GbE/2.5GbE Multilayer Ethernet Switch

Table 102: Switch Devices

Family	Devices	Description
	BCM53416 A0	12-port 10GbE plus 8-port 2.5GbE and 4-port 5GbE/2.5GbE Ethernet Switch with integrated CPU
BCM53600	BCM53602 A0	8-Port Fast Ethernet + 3-Port Gigabit Ethernet Switch with one 1/2G-EPON ONU MAC/SerDes and embedded 600MHz MIPS32 74K processor
	BCM53603 A0	16-Port Fast Ethernet + 3-Port Gigabit Ethernet Switch with one 1/2G-EPON ONU MAC/SerDes and embedded 600MHz MIPS32 74K processor
	BCM53604 A0	24-Port Fast Ethernet + 3-Port Gigabit Ethernet Switch with one 1/2G-EPON ONU MAC/SerDes and embedded 600MHz MIPS32 74K processor
	BCM53606 A0	24-Port FE with S3MII interface + 3-Port Gigabit Ethernet Switch with one 1/2G-EPON ONU MAC/SerDes and embedded 600MHz MIPS32 74K processor
BCM89500	BCM89500 A0	4-Port Integrated Dedicated BRPHY + 3-Port Gigabit Ethernet Switch with embedded ARM processor
	BCM89500 B0	
BCM89500	BCM89501 A0	4-Port Integrated Dedicated BRPHY + 1-Port Integrated Dual-Mode BRPHY + 2-Port Gigabit Ethernet Switch with embedded ARM processor
	BCM89501 B0	
BCM89500	BCM89200 A0	1-Port Integrated Dedicated BRPHY + 1-Port Integrated Dual-Mode BRPHY + 2-Port Gigabit Ethernet Switch with embedded ARM processor
	BCM89200 B0	
BCM53710	BCM53714 A0	BCM56714 Integrated Multilayer Switch and CPU
	BCM53714 A1	
	BCM53714 A2	
	BCM53716 A0	BCM56716 Integrated Multilayer Switch and CPU
	BCM53716 A1	
	BCM53716 A2	
	BCM53718 A0	BCM56718 Integrated Multilayer Switch and CPU
	BCM53718 A1	
	BCM53718 A2	
BCM53720	BCM53724 A0	Managed 24-port L2 Switch with Integrated CPU
	BCM53724 B0	
	BCM53726 A0	Managed 24-port L2 Switch with Integrated CPU
	BCM53726 B0	
	BCM5675 A1	
	BCM5676 A0	4-Port, 96-Gbps Switch Fabric
	BCM5676 A1	
BCM56010	BCM56014 A0	24-Port Integrated Multilayer Switch and CPU
	BCM56014 A1	
	BCM56014 A2	
	BCM56018 A0	48-Port Integrated Multilayer Switch and CPU
	BCM56018 A1	
	BCM56018 A2	
	BCM56018 A1	48-Port Integrated Multilayer Switch and CPU
BCM56020	BCM56024 A0	24-Port Integrated Multilayer Switch and CPU
	BCM56024 B0	

Table 102: Switch Devices

Family	Devices	Description
	BCM56025 A0	24-Port Integrated L2 Switch and CPU
	BCM56025 B0	
	BCM56026 A0	24-Port Integrated L2 Switch and CPU
	BCM56026 B0	
BCM56060	BCM56060 A0	16-port 10GbE Multilayer Ethernet Switch with integrated CPU
	BCM56063 A0	16-port 1GbE plus 4-port 10GbE (XFI) Multilayer Switch with integrated CPU
	BCM56064 A0	24-port GbE plus 4-port 10GbE Multilayer Managed Switch with HiGi Uplinks and integrated CPU
BCM56100	BCM56100 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch
	BCM56100 A1	
	BCM56101 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch with One 10-Gigabit Ethernet/HiGig Port
	BCM56101 A1	
	BCM56102 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch with Two 10-Gigabit Ethernet/HiGig Ports
	BCM56102 A1	
	BCM56105 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch
	BCM56105 A1	
	BCM56106 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch with One 10-Gigabit Ethernet/HiGig Port
	BCM56106 A1	
	BCM56107 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch with Two 10-Gigabit Ethernet/HiGig Ports
	BCM56107 A1	
BCM56110	BCM56110 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch
	BCM56111 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch with One 10-Gigabit Ethernet/HiGig Port
	BCM56112 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch with Two 10-Gigabit Ethernet/HiGig Ports
	BCM56115 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch
	BCM56116 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch with One 10-Gigabit Ethernet/HiGig Port
	BCM56117 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch with Two 10-Gigabit Ethernet/HiGig Ports
BCM56130	BCM56132 A0	24-Port Fast Ethernet Multilayer Switch with Two 10-GbE/HiGig2 and Two 1G/2.5Gb Uplink Ports
	BCM56132 B0	
	BCM56132 B1	
	BCM56134 A0	24-Port Fast Ethernet Multilayer Switch with four 1G/2.5Gb Uplink Ports
	BCM56134 B0	
	BCM56134 B1	
BCM56140	BCM56140 A0	24-Port Gigabit Ethernet/6-Port SGMII GbE Multilayer switch with combination of two/four 1G/2.5/HiGig2 Uplink Ports
	BCM56142 A0	24-Port Gigabit Ethernet Multilayer switch with combination of two/four 1G/2.5/HiGig2 Uplink Ports
	BCM56143 A0	24-Port Gigabit Ethernet Multilayer switch with combination of two/four 1G/2.5/HiGig2 Uplink Ports
	BCM56144 A0	16-Port Gigabit Ethernet Multilayer switch with four 1G/2.5HG Uplink Ports

Table 102: Switch Devices

Family	Devices	Description
	BCM56146 A0	24-Port Fast-Ethernet Multilayer switch with four 2.5HG Uplink Ports
	BCM56147 A0	24-Port Fast-Ethernet Multilayer switch with combination of one/two/four 1G/2.5G/10/12/13HG Uplink Ports
BCM56150	BCM56150 A0	24-port GbE Managed Switch with 4-port 10 GbE uplinks, integrated CPU and 16 copper PHYs
	BCM56151 A0	24-port GbE Managed Switch with 4-port 10 GbE uplinks, integrated CPU (without PHYs)
	BCM56152 A0	24-port GbE plus 2-port GbE and 2-port 1GbE/13GbE uplinks Managed Switch, integrated CPU and 16 copper PHYs
	BCM53342 A0	8-port GbE Multilayer WebSmart Switch with Integrated CPU and Copper PHYs
	BCM53343 A0	16-port GbE plus 4-port GbE uplinks Multilayer WebSmart Switch with Integrated CPU and 16 Copper PHYs
	BCM53344 A0	24-port GbE plus 2-port GbE and 2-port 1GbE/13GbE uplinks WebSmart Switch, integrated CPU and 16 copper PHYs
	BCM53346 A0	24-port GbE Multilayer WebSmart Switch with 4-port 10 GbE uplinks, integrated CPU and 16 copper PHYs
	BCM53347 A0	24-port GbE Multilayer WebSmart Switch with 6xQSGMII + 4x1/10G
	BCM53393 A0	14-port GbE Multilayer Embedded Switch with integrated CPU (without PHY)
	BCM53394 A0	10-port GbE Multilayer Embedded Switch with 4-port 10 GbE uplinks, integrated CPU (without PHY)
BCM56210	BCM56212 A0	
	BCM56212 A1	
	BCM56212 A2	
	BCM56213 A0	
	BCM56213 A1	
	BCM56213 A2	
	BCM56214 A0	BCM56214 Integrated Multilayer Switch and CPU
	BCM56214 A1	
	BCM56214 A2	
	BCM56215 A0	
	BCM56215 A1	
	BCM56215 A2	
	BCM56216 A0	BCM56216 Integrated Multilayer Switch and CPU
	BCM56216 A1	
	BCM56216 A2	
	BCM56217 A0	
	BCM56217 A1	
	BCM56217 A2	
	BCM56218 A0	BCM56218 Integrated Multilayer Switch and CPU
	BCM56218 A1	
	BCM56218 A2	
	BCM56219 A0	BCM56219 Integrated Multilayer Switch and CPU
	BCM56219 A1	
	BCM56219 A2	
BCM56220	BCM56224 A0	24 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+

Table 102: Switch Devices

Family	Devices	Description
	BCM56224 B0	24 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56225 A0	24 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56225 B0	24 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56226 A0	16 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56226 B0	16 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56227 A0	16 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56227 B0	16 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56228 A0	8 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56228 B0	8 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56229 A0	8 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56229 B0	8 GbE + 4 x 1 Gb/2.5 Gb, L2+
BCM56230	BCM56230 B1	12-Port GbE Multilayer Switch
	BCM56231 B1	6-Port GbE Multilayer Switch
BCM56300	BCM56300 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56300 A1	
	BCM56300 B0	
	BCM56300 B1	
	BCM56301 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56301 A1	
	BCM56301 B0	
	BCM56301 B1	
	BCM56302 A0	24-Port Gigabit Ethernet Multilayer Switch with Two 10-Gigabit Ethernet/ HiGig+ Ports
	BCM56302 A1	
	BCM56302 B0	
	BCM56302 B1	
	BCM56303 A0	24-Port Gigabit Ethernet Multilayer Switch with Three 10 Gigabit Ethernet/ HiGig+ Ports
	BCM56303 A1	
	BCM56303 B0	
	BCM56303 B1	
	BCM56304 A0	24-Port Gigabit Ethernet Multilayer Switch with Four 10-Gigabit Ethernet/ HiGig+ Ports
	BCM56304 A1	
	BCM56304 B0	
	BCM56304 B1	
	BCM56305 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56305 A1	
	BCM56305 B0	
	BCM56305 B1	
	BCM56306 A0	16 Port Gigabit Ethernet Switch
	BCM56306 A1	
	BCM56306 B0	

Table 102: Switch Devices

Family	*Devices*	*Description*
	BCM56306 B1	
	BCM56307 A0	24-Port GE L2 Switch with Two 10 GE/HiGig+ Ports
	BCM56307 A1	
	BCM56307 B0	
	BCM56307 B1	
	BCM56308 A0	24-Port GE L2 Switch with Three 10 GE/HiGig+ Ports
	BCM56308 A1	
	BCM56308 B0	
	BCM56308 B1	
	BCM56309 A0	24-Port GE L2 Switch with Four 10 GE/HiGig+ Ports
	BCM56309 A1	
	BCM56309 B0	
	BCM56309 B1	
BCM56310	BCM56310 A0	BCM56310 Series 24-Port GbE Multilayer Switch with Four 10-GbE/HiGig+ Uplink Ports
	BCM56311 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56312 A0	24-Port Gigabit Ethernet Multilayer Switch with Two 10-Gigabit Ethernet/ HiGig+ Ports
	BCM56313 A0	24-Port Gigabit Ethernet Multilayer Switch with Three 10-Gigabit Ethernet/ HiGig+ Ports
	BCM56314 A0	24-Port Gigabit Ethernet Multilayer Switch with Four 10-Gigabit Ethernet/ HiGig+ Ports
	BCM56315 A0	BCM56310 Series 24-Port GbE Layer 2 Switch with Four 10-GbE/HiGig+ Uplink Ports
	BCM56316 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56317 A0	24-Port Gigabit Ethernet Layer 2 Switch with Two 10-Gigabit Ethernet/HiGig+ Ports
	BCM56318 A0	24-Port Gigabit Ethernet Layer 2 Switch with Three 10-Gigabit Ethernet/ HiGig+ Ports
	BCM56319 A0	24-Port Gigabit Ethernet Layer 2 Switch with Four 10-Gigabit Ethernet/HiGig+ Ports
BCM56320	BCM56320 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56320 B0	
	BCM56320 B1	
	BCM56321 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56321 B0	
	BCM56321 B1	
BCM56330	BCM56331 A0	24-Port GbE Multilayer Switch with Four 2.5GbE Uplink Ports
	BCM56331 B0	
	BCM56331 B1	
	BCM56333 A0	16-Port GbE Multilayer Switch
	BCM56333 B0	
	BCM56333 B1	
	BCM56334 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56334 B0	

Table 102: Switch Devices

Family	Devices	Description
	BCM56334 B1	
	BCM56338 A0	8-Port GbE Multilayer Switch with two 10-GbE/HiGig2 Uplink Ports
	BCM56338 B0	
	BCM56338 B1	
BCM56340	BCM56040 A0	1xF.QSGMII + 3xF.HG[42] + 1GE
	BCM56041 A0	Ranger device, meant for embedded connectivity supports 1Ge (port 49), 2 X GE (iPROC), Flex 4x10G, 3 X 4 X 10G
	BCM56042 A0	12x2.5GE/1GE + 12x2.5GE/1GE + 1GE
	BCM56340 A0	12xF.QSGMII + Flex[4x10] + 2xHG[21] + 1GE, 12xF.QSGMII + 4xSGMII + 2xXFI + 2xHGd[21] + 1GE
	BCM56342 A0	7xF.QSGMII + Flex[4x10] + 2xHG[21] + 1GE
	BCM56344 A0	10xF.QSGMII + 3xFlex[4x10] + 1GE
BCM56340	BCM56547 A0	10xF.QSGMII + 3xF.HG[42] + 1GE, 12xF.QSGMII + 2xF.HG[42] + 1GE, 12xF.QSGMII + F.HG[42] + 2xHG[42] + 1GE
BCM56340	BCM56548 A0	7xF.QSGMII + 3xF.HG[42] + 1GE
BCM56440	BCM56440 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports
	BCM56440 B0	
	BCM56441 A0	8-Port GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56441 B0	
	BCM56442 A0	16-Port GbE Multilayer Switch
	BCM56442 B0	
	BCM56443 A0	8-Port 2.5GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56443 B0	
	BCM56445 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports pin compatible with BCM56334
	BCM56445 B0	
	BCM56446 A0	8-Port GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports pin compatible with BCM56338
	BCM56447 A0	16-Port GbE Multilayer Switch pin compatible with BCM56333
	BCM56447 B0	
	BCM56448 A0	24-Port GbE Multilayer Switch with Four 1GbE/ One 2.5G Uplink ports
	BCM56448 B0	
BCM56450	BCM56248L B0	11xGE + 8x2.5G
BCM56450	BCM56450 A0	24-port GbE Multilayer Switch with 4-port 10 GbE uplinks, stacking, integrated CPU and Traffic Manager
BCM56450	BCM56450 B0	Katana2 Access 1 x XAUI + 8 x GE without L3 routing and MPLS features
BCM56450	BCM55450 B0	KT2 Access-8 FX + 2 F-HG
	BCM56455 A0	2 x 20GE (G.INT) + 2 x HG13
	BCM56455 B0	2 x 20GE (G.INT) + 2 x HG13
BCM56450	BCM56456 A0	24x GE + 4x F.XAUI
BCM56450	BCM56456 B0	24x GE + 4x F.XAUI
	BCM56456 B0	1 x XAUI + 8 x GE
BCM56456	BCM56456 B0	9xFXAUI + 1 x XAUI + 1x2.5GbE
BCM56456	BCM56458 B0	8xGE + 2xF.XAUI
BCM56500	BCM56500 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56500 A1	

Table 102: Switch Devices

Family	*Devices*	*Description*
	BCM56500 B0	
	BCM56500 B1	
	BCM56500 B2	
	BCM56501 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56501 A1	
	BCM56501 B0	
	BCM56501 B1	
	BCM56501 B2	
	BCM56502 A0	24-Port GbE Multilayer Switch with Two 10-GbE/HiGig+ Ports
	BCM56502 A1	
	BCM56502 B0	
	BCM56502 B1	
	BCM56502 B2	
	BCM56503 A0	24-Port GbE Multilayer Switch with Three 10-GbE/HiGig+ Ports
	BCM56503 A1	
	BCM56503 B0	
	BCM56503 B1	
	BCM56503 B2	
	BCM56504 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig+ Ports
	BCM56504 A1	
	BCM56504 B0	
	BCM56504 B1	
	BCM56504 B2	
	BCM56505 A0	24-Port GbE Layer 2 Switch
	BCM56505 A1	
	BCM56505 B0	
	BCM56505 B1	
	BCM56505 B2	
	BCM56506 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56506 A1	
	BCM56506 B0	
	BCM56506 B1	
	BCM56506 B2	
	BCM56507 A0	24-Port GbE Layer 2 Switch with Two 10-GbE/HiGig+ Ports
	BCM56507 A1	
	BCM56507 B0	
	BCM56507 B1	
	BCM56507 B2	
	BCM56508 A0	24-Port GbE Layer 2 Switch with Three 10-GbE/HiGig+ Ports
	BCM56508 A1	
	BCM56508 B0	

Table 102: Switch Devices

Family	Devices	Description
	BCM56508 B1	
	BCM56508 B2	
	BCM56509 A0	24-Port GbE Layer 2 Switch with Four 10-GbE/HiGig+ Ports
	BCM56509 A1	
	BCM56509 B0	
	BCM56509 B1	
	BCM56509 B2	
BCM56510	BCM56510 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56511 A0	Four-Port 10-GbE/HiGig+ Multilayer Switch
	BCM56512 A0	24-Port GbE Multilayer Switch With Two 10-GbE/HiGig+ Ports
	BCM56513 A0	24-Port GbE Multilayer Switch With Three 10-GbE/HiGig+ Ports
	BCM56514 A0	24-Port GbE Multilayer Switch With Four 10-GbE/HiGig+ Ports
BCM56520	BCM56520 A0	24-Port GbE Multilayer Switch
	BCM56520 B0	
	BCM56522 A0	24-Port GbE Multilayer Switch with Two 10-GbE/HiGig2 Uplink Ports
	BCM56522 B0	
	BCM56524 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56524 B0	
	BCM56526 A0	28-Port GbE Multilayer Switch with Six 10-GbE/HiGig2 Uplink Ports
	BCM56526 B0	
BCM56530	BCM56534 B0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56538 B0	48-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
BCM56540	BCM56540 A1	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56540 B0	
	BCM56541 A1	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56541 B0	
	BCM56542 A1	28xGE + 2xF.XAUI/2x10GE + 2xF.HG[42] + 2xF.HG[21] + 1GE, 28xGE + 8xGE/8x2.5GE + 2xHG[42] + 2xHG[21] + 1GE Multilayer Ethernet Switch
	BCM56544 A1	10xF.XAUI + 4xHG[21] + 1GE, 10xF.XAUI + 4xXFI, 10xF.XAUI + 2xHG[42], 4xXAUI + 12xXFI + 1GE Multilayer Ethernet Switch
	BCM56544 B0	10xF.XAUI + 4xHG[21] + 1GE, 10xF.XAUI + 4xXFI, 10xF.XAUI + 2xHG[42], 4xXAUI + 12xXFI + 1GE Multilayer Ethernet Switch
	BCM56545 A1	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56545 B0	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE, 24xGE + 4xXAUI + 2xXFI + 2xHG[12] + 1GE Multilayer Ethernet Switch
	BCM56546 A1	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56546 B0	
BCM56580	BCM56580 A0	16 x 2.5 GbE + 4 x 10 GbE Ethernet Multilayer Switch
BCM56620	BCM56620 A0	
	BCM56620 A1	
	BCM56620 B0	

Table 102: Switch Devices

Family	*Devices*	*Description*
	BCM56620 B1	
	BCM56620 B2	
	BCM56624 A0	49 port 1-GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56624 A1	
	BCM56624 B0	
	BCM56624 B1	
	BCM56624 B2	
	BCM56626 A0	25 port 1-GbE Multilayer Ethernet Switch with 6 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56626 A1	
	BCM56626 B0	
	BCM56626 B1	
	BCM56626 B2	
	BCM56628 A0	8 port 10-GbE/HiGig2 Multilayer Ethernet Switch with External Table Expansion
	BCM56628 A1	
	BCM56628 B0	
	BCM56628 B1	
	BCM56628 B2	
	BCM56629 B0	25 port 1-GbE Multilayer Ethernet Switch with 8 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56629 B1	
	BCM56629 B2	
BCM56630	BCM56630 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56630 B0	
	BCM56634 A0	48-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56634 B0	
	BCM56636 A0	24-Port GbE + 2-Port 10-GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56636 B0	
	BCM56638 A0	4-Port 10-GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56638 B0	
	BCM56639 A0	24-Port GbE + 4-Port 10-GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56639 B0	
BCM56640	BCM56045 B0	3xF.40GE + 3xF.HG[42] + 1GE
	BCM56046 B0	3xF.40GE + 2xF.HG[42] + 1GE
	BCM56640 A1	1x100GE + 1xHG[127], 1x100GE + 4xHG[32], 1x100GE + 8xHGd[16], 3xF.HG[42] + 1xHG[127], 3xF.HG[42] + 4xHG[32], 3xF.HG[42] + 8xHGd[16], 3xF.HG[42] + 3xF.HG[42] Multilayer Ethernet Switch
	BCM56640 B0	
	BCM56643 A1	48xGE + 4xXFI + 4xHG[42] + 1GE Multilayer Ethernet Switch
	BCM56643 B0	
	BCM56644 A1	48xGE + 2xHG[25] + 2xHG[25] + 1GE Multilayer Ethernet Switch

Table 102: Switch Devices

Family	*Devices*	*Description*
	BCM56644 B0	
	BCM56648 A1	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56648 B0	
	BCM56649 A1	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56649 B0	
BCM56680	BCM56680 A0	25 port 1-GbE/2.5GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports
	BCM56680 A1	
	BCM56680 B0	
	BCM56680 B1	
	BCM56684 A0	24 port 1-GbE/2.5GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports
	BCM56684 A1	
	BCM56684 B0	
	BCM56684 B1	
BCM56685	BCM56685 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56685 B0	
	BCM56689 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56689 B0	
BCM56700	BCM56700 A0	16-Port, 192-Gbps Lossless Switch Fabric
	BCM56701 A0	12-Port, 144-Gbps Lossless Switch Fabric
BCM56720	BCM56720 A0	16 Port, 16-Gbps HiGig2 Switch Fabric
	BCM56721 A0	12 Port, 16-Gbps HiGig2 Switch Fabric
BCM56725	BCM56725 A0	8 Port, 20-Gbps + 4 Port, 16-Gbps HiGig2 Switch Fabric
BCM56740	BCM56743 A0	480 Gbps Switch fabric
	BCM56743 A1	
	BCM56743 A2	
	BCM56743 A3	
	BCM56743 A4	
	BCM56743 B0	
	BCM56743 B1	
	BCM56745 A0	640 Gbps Switch fabric
	BCM56745 A1	
	BCM56745 A2	
	BCM56745 A3	
	BCM56745 A4	
	BCM56745 B0	
	BCM56745 B1	
BCM56740_PLUS	BCM56744 A0	480 Gbps Switch fabric
	BCM56744 A1	
	BCM56746 A0	640 Gbps Switch fabric
	BCM56746 A1	

Table 102: Switch Devices

Family	Devices	Description
BCM56800	BCM56800 A0	20-Port 10-Gigabit Ethernet Multilayer Switch
	BCM56801 A0	10-Port 10-Gigabit Ethernet and 8-Port HiGig2/10GbE Multilayer Switch
	BCM56802 A0	16-Port 10-GbE/HiGig2 Multilayer Switch
	BCM56803 A0	12 Port 10GE/HiGig2 Multilayer Switch
BCM56820	BCM56820 A0	24 x 10-GbE + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56820 B0	
	BCM56821 A0	12 x 10-GbE + 8 x HiGig2 + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56821 B0	
	BCM56822 A0	12 x 10-GbE + 4 x 20-Gbps HiGig2 + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56822 B0	
	BCM56823 A0	8 x 10-GbE + 4 x 20-Gbps HiGig2 + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56823 B0	
	BCM56825 B0	16 x 10-GbE + 8 x 20-Gbps HiGig2 + 1 x 1-GbE Multilayer Ethernet Switch
BCM56740	BCM56743 A0	480 Gbps Switch fabric
	BCM56743 A1	
	BCM56743 A2	
	BCM56743 A3	
	BCM56743 A4	
	BCM56743 B0	
	BCM56743 B1	
	BCM56745 A0	640 Gbps Switch fabric
	BCM56745 A1	
	BCM56745 A2	
	BCM56745 A3	
	BCM56745 A4	
	BCM56745 B0	
	BCM56745 B1	
BCM56740_PLUS	BCM56744 A0	480 Gbps Switch fabric
	BCM56744 A1	
	BCM56746 A0	640 Gbps Switch fabric
	BCM56746 A1	
BCM56840	BCM56841 A0	320 Gbps Ethernet Multilayer Switch
	BCM56841 A1	
	BCM56841 A2	
	BCM56841 A3	
	BCM56841 A4	
	BCM56841 B0	
	BCM56841 B1	
	BCM56843 A0	480 Gbps Ethernet Multilayer Switch
	BCM56843 A1	
	BCM56843 A2	
	BCM56843 A3	

Table 102: Switch Devices

Family	Devices	Description
	BCM56843 A4	
	BCM56843 B0	
	BCM56843 B1	
	BCM56845 A0	640 Gbps Ethernet Multilayer Switch
	BCM56845 A1	
	BCM56845 A2	
	BCM56845 A3	
	BCM56845 A4	
	BCM56845 B0	
	BCM56845 B1	
BCM56840_PLUS	BCM56842 A0	320 Gbps Ethernet Multilayer Switch
	BCM56842 A1	
	BCM56844 A0	480 Gbps Ethernet Multilayer Switch
	BCM56844 A1	
	BCM56846 A0	640 Gbps Ethernet Multilayer Switch
	BCM56846 A1	
BCM56846	BCM56831	Trident+ SKU - 24-port 10GE switch with 40GE support for embedded applications
BCM56846	BCM56835C	Trident+ SKU - (64 x 10 GbE) + (4 x 1 GbE)
BCM56846	BCM56847	Trident+ SKU - (64 x 10 GbE) + (4 x 1 GbE)
BCM56846	BCM56849	Trident+ SKU - (56 x 1GbE/2.5GbE) + (8 x 10GbE)
BCM56850	BCM56751P A1	1.28Tbps I/O, 960Gbps Core Ethernet Switch Fabric
BCM56850	BCM56751P A2	1.28Tbps I/O, 960Gbps Core Ethernet Switch Fabric
BCM56850	BCM56830 A1	960Gbps Ethernet Switch
BCM56850	BCM56830 A2	960Gbps Ethernet Switch
BCM56850	BCM56834	Trident2 SKU - High density 10G and 40G switch for embedded applications
BCM56850	BCM56838	Trident2 SKU - 72/320G Devices with 1.25/3.125/6.25G Serdes and 4 SFIs
BCM56850	BCM56850 A1	1.28Tbps I/O, 1Tbps Core Ethernet Switch
	BCM56852 A2	100x10G, 960Gbps Multilayer Switch
	BCM56854 A1	
BCM56851	BCM56751 A2	1.28Tbps I/O, 960Gbps Core Ethernet Switch Fabric
BCM88732	BCM88732 B2	Eight-Port 10 GbE or 2-Port 40 GbE MAC Aggregation Switch with 80 Gbps Uplink Capacity
BCM88020	BCM88020 A0	XGS Core (XCore/SBX) Fully Programmable Carrier Packet Processor with 24 GbE Ports, 2 10GbE Ports and 2 SPI Interfaces
	BCM88020 A1	
	BCM88020 A2	
BCM88025	BCM88025 A0	XGS Core (XCore/SBX) Fully Programmable Carrier Packet Processor with 24 GbE Ports, 2 10GbE Ports and 2 SPI Interfaces
BCM88030	BCM88030 A0	XGS Core (XCore/SBX) Scalable Switching 100 Gbps Fully Programmable Carrier Packet Processor
BCM88130	BCM88130 A0	XGS Core (XCore/SBX) 630 Gbps Bandwidth Manager and Switching Engine
	BCM88130 A1	
BME-3200	BME-3200 A0	XGS Core (XCore/SBX) Fabric Bandwidth Manager with 32 SCI control ports and up to 40 SFI data ports
	BME-3200 B0	
QE-2000	QE-2000 A1	XGS Core (XCore/SBX) Fabric Queuing Engine with 49 SPI 4.2 subports

Table 102: Switch Devices

Family	Devices	Description
	QE-2000 A2	
	QE-2000 A3	
	QE-2000 A4	
BCM88230	BCM88230 A0	XGS Core (XCore/SBX) Fabric Queuing Engine with Integrated Traffic Management with 4 HiGig2 ports, 50Gbps
	BCM88230 B0	
	BCM88235 A0	XGS Core (XCore/SBX) Fabric Queuing Engine with Integrated Traffic Management with 4 HiGig2 ports, 80Gbps
	BCM88235 B0	
	BCM88231 A0	XGS Core (XCore/SBX) Traffic Manager with 4 HiGig2 ports, 50Gbps
	BCM88231 B0	
	BCM88236 A0	XGS Core (XCore/SBX) Traffic Manager with 4 HiGig2 ports, 80Gbps
	BCM88236 B0	
BCM56930	BCM56931 A0	XGS pass-through and standalone Traffic Manager, 4 HiGig2 ports, 50Gbps
	BCM56931 B0	
	BCM56936 A0	XGS pass-through and standalone Traffic Manager, 4 HiGig2 ports, 80Gbps
	BCM56936 B0	
BCM88640	BCM88640 A0	DNX 100G Flexible Packet Processor with Integrated Traffic Management
	BCM88640 B0	
BCM88650	BCM88650 A0	DNX 200G Flexible Packet Processor with Integrated Traffic Management
	BCM88650 B0	
	BCM88650 B1	200 GBps DNX Traffic Manager and Packet Processor
BCM88660	BCM88660 A0	DNX 200G Flexible Packet Processor with Integrated Traffic Management
BCM88750	BCM88750 A0	DNX 1600 GBps Switch Fabric
	BCM88750 B0	

Table 103: SER Supported Devices

Family	Devices
Trident	56841, 56842, 56843, 56844, 56845, 56846, 56850
Triumph	56640, 56643, 56644, 56648, 56649, 56540, 56541, 56542, 56544, 56545
Katana	All SKUs
Katana2	56450, 56455, 56456
Enduro2	All SKUs
Hurricane2	56150, 56151, 53344, 53346, 53393, 53394
Helix4	56340, 56040, 56344, 56042, 56342

Table 104: Switch Devices that support Warm boot

Family	Devices	Description
BCM5675	BCM5675 A0	8-Port, 192-Gbps Switch Fabric
	BCM5675 A1	

Table 104: Switch Devices that support Warm boot

Family	Devices	Description
	BCM5676 A0	4-Port, 96-Gbps Switch Fabric
	BCM5676 A1	
BCM56020	BCM56024 A0	24-Port Integrated Multilayer Switch and CPU
	BCM56024 B0	
	BCM56025 A0	24-Port Integrated L2 Switch and CPU
	BCM56025 B0	
	BCM56026 A0	24-Port Integrated L2 Switch and CPU
	BCM56026 B0	
BCM56130	BCM56132 A0	24-Port Fast Ethernet Multilayer Switch with Two 10-GbE/HiGig2 and Two 1G/2.5Gb Uplink Ports
	BCM56132 B0	
	BCM56132 B1	
	BCM56134 A0	24-Port Fast Ethernet Multilayer Switch with four 1G/2.5Gb Uplink Ports
	BCM56134 B0	
	BCM56134 B1	
BCM56142	BCM56142 A0	24-Port Fast Ethernet Multilayer Switch with four 1G/2.5Gb/Higig2/HG Lite Uplink Ports
BCM56150	BCM56150 A0	24-port GbE Managed Switch with 4-port 10 GbE uplinks, integrated CPU and 16 copper PHYs
	BCM56151 A0	24-port GbE Managed Switch with 4-port 10 GbE uplinks, integrated CPU (without PHYs)
	BCM56152 A0	24-port GbE plus 2-port GbE and 2-port 1GbE/13GbE uplinks Managed Switch, integrated CPU and 16 copper PHYs
	BCM53342 A0	8-port GbE Multilayer WebSmart Switch with Integrated CPU and Copper PHYs
	BCM53343 A0	16-port GbE plus 4-port GbE uplinks Multilayer WebSmart Switch with Integrated CPU and 16 Copper PHYs
	BCM53344 A0	24-port GbE plus 2-port GbE and 2-port 1GbE/13GbE uplinks WebSmart Switch, integrated CPU and 16 copper PHYs
	BCM53346 A0	24-port GbE Multilayer WebSmart Switch with 4-port 10 GbE uplinks, integrated CPU and 16 copper PHYs
	BCM53393 A0	14-port GbE Multilayer Embedded Switch with integrated CPU (without PHY)
	BCM53394 A0	10-port GbE Multilayer Embedded Switch with 4-port 10 GbE uplinks, integrated CPU (without PHY)
BCM56220	BCM56224 A0	24 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56224 B0	24 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56225 A0	24 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56225 B0	24 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56226 A0	16 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56226 B0	16 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56227 A0	16 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56227 B0	16 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56228 A0	8 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56228 B0	8 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56229 A0	8 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56229 B0	8 GbE + 4 x 1 Gb/2.5 Gb, L2+
BCM56230	BCM56230 B1	12-Port GbE Multilayer Switch

Table 104: Switch Devices that support Warm boot

Family	Devices	Description
	BCM56231 B1	6-Port GbE Multilayer Switch
BCM56240	BCM56240 A0	2-Port 10GbE (OR 8 *2.5GbE) Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
BCM56240	BCM56240 B0	2-Port 10GbE (OR 8 *2.5GbE) Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56241 A0	6-Port GbE Multilayer Switch with Two 2.5GbE Uplink ports
	BCM56242 A0	8-Port 2.5GbE Multilayer Switch with Two 2.5GbE Uplink ports
	BCM56243 A0	4-Port 2.5GbE Multilayer Switch
BCM56320	BCM56320 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56320 B0	
	BCM56320 B1	
	BCM56321 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56321 B0	
	BCM56321 B1	
BCM56330	BCM56331 A0	24-Port GbE Multilayer Switch with Four 2.5GbE Uplink Ports
	BCM56331 B0	
	BCM56331 B1	
	BCM56333 A0	16-Port GbE Multilayer Switch
	BCM56333 B0	
	BCM56333 B1	
	BCM56334 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56334 B0	
	BCM56334 B1	
	BCM56338 A0	8-Port GbE Multilayer Switch with two 10-GbE/HiGig2 Uplink Ports
	BCM56338 B0	
	BCM56338 B1	
BCM56340	BCM56340 A0	12xF.QSGMII + Flex[4x10] + 2xHG[21] + 1GE, 12xF.QSGMII + 4xSGMII + 2xXFI + 2xHGd[21] + 1GE
	BCM56342 A0	7xF.QSGMII + Flex[4x10] + 2xHG[21] + 1GE
	BCM56344 A0	10xF.QSGMII + 3xFlex[4x10] + 1GE
	BCM56040 A0	1xF.QSGMII + 3xF.HG[42] + 1GE
	BCM56041 A0	Ranger device, meant for embedded connectivity supports 1Ge (port 49), 2 X GE (iPROC), Flex 4x10G, 3 X 4 X 10G
	BCM560547 A0	10xF.QSGMII + 3xF.HG[42] + 1GE, 12xF.QSGMII + 2xF.HG[42] + 1GE, 12xF.QSGMII + F.HG[42] + 2xHG[42] + 1GE
	BCM560548 A0	7xF.QSGMII + 3xF.HG[42] + 1GE
BCM56240	BCM56240 A0	2-Port 10GbE (OR 8 *2.5GbE) Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
BCM56240	BCM56240 B0	2-Port 10GbE (OR 8 *2.5GbE) Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56241 A0	6-Port GbE Multilayer Switch with Two 2.5GbE Uplink ports
	BCM56242 A0	8-Port 2.5GbE Multilayer Switch with Two 2.5GbE Uplink ports
	BCM56243 A0	4-Port 2.5GbE Multilayer Switch
BCM56440	BCM55441 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports
	BCM56440 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports

Table 104: Switch Devices that support Warm boot

Family	Devices	Description
	BCM56440 B0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports
	BCM56441 A0	8-Port GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56442 A0	16-Port GbE Multilayer Switch
	BCM56443 A0	8-Port 2.5GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56445 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports pin compatible with BCM56334
	BCM56446 A0	8-Port GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports pin compatible with BCM56338
	BCM56447 A0	16-Port GbE Multilayer Switch pin compatible with BCM56333
	BCM56448 A0	24-Port GbE Multilayer Switch with Four 1GbE/ One 2.5G Uplink ports
BCM56450	BCM56450 A0	24-port GbE Multilayer Switch with 4-port 10 GbE uplinks, stacking, integrated CPU and Traffic Manager
	BCM56455 A0	2 x 20GE (G.INT) + 2 x HG13
	BCM56456 B0	1 x XAUI + 8 x GE
BCM56500	BCM56500 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56500 A1	
	BCM56500 B0	
	BCM56500 B1	
	BCM56500 B2	
	BCM56501 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56501 A1	
	BCM56501 B0	
	BCM56501 B1	
	BCM56501 B2	
	BCM56502 A0	24-Port GbE Multilayer Switch with Two 10-GbE/HiGig+ Ports
	BCM56502 A1	
	BCM56502 B0	
	BCM56502 B1	
	BCM56502 B2	
	BCM56503 A0	24-Port GbE Multilayer Switch with Three 10-GbE/HiGig+ Ports
	BCM56503 A1	
	BCM56503 B0	
	BCM56503 B1	
	BCM56503 B2	
	BCM56504 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig+ Ports
	BCM56504 A1	
	BCM56504 B0	
	BCM56504 B1	
	BCM56504 B2	
	BCM56505 A0	24-Port GbE Layer 2 Switch
	BCM56505 A1	
	BCM56505 B0	

Table 104: Switch Devices that support Warm boot

Family	Devices	Description
	BCM56505 B1	
	BCM56505 B2	
	BCM56506 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56506 A1	
	BCM56506 B0	
	BCM56506 B1	
	BCM56506 B2	
	BCM56507 A0	24-Port GbE Layer 2 Switch with Two 10-GbE/HiGig+ Ports
	BCM56507 A1	
	BCM56507 B0	
	BCM56507 B1	
	BCM56507 B2	
	BCM56508 A0	24-Port GbE Layer 2 Switch with Three 10-GbE/HiGig+ Ports
	BCM56508 A1	
	BCM56508 B0	
	BCM56508 B1	
	BCM56508 B2	
	BCM56509 A0	24-Port GbE Layer 2 Switch with Four 10-GbE/HiGig+ Ports
	BCM56509 A1	
	BCM56509 B0	
	BCM56509 B1	
	BCM56509 B2	
BCM56510	BCM56510 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56511 A0	Four-Port 10-GbE/HiGig+ Multilayer Switch
	BCM56512 A0	24-Port GbE Multilayer Switch With Two 10-GbE/HiGig+ Ports
	BCM56513 A0	24-Port GbE Multilayer Switch With Three 10-GbE/HiGig+ Ports
	BCM56514 A0	24-Port GbE Multilayer Switch With Four 10-GbE/HiGig+ Ports
BCM56520	BCM56520 A0	24-Port GbE Multilayer Switch
	BCM56520 B0	
	BCM56522 A0	24-Port GbE Multilayer Switch with Two 10-GbE/HiGig2 Uplink Ports
	BCM56522 B0	
	BCM56524 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56524 B0	
	BCM56526 A0	28-Port GbE Multilayer Switch with Six 10-GbE/HiGig2 Uplink Ports
	BCM56526 B0	
BCM56530	BCM56534 B0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56538 B0	48-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
BCM56620	BCM56620 A0	
	BCM56620 A1	
	BCM56620 B0	
	BCM56620 B1	

Table 104: Switch Devices that support Warm boot

Family	Devices	Description
	BCM56624 A0	49 port 1-GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56624 A1	
	BCM56624 B0	
	BCM56624 B1	
	BCM56624 B2	
	BCM56626 A0	25 port 1-GbE Multilayer Ethernet Switch with 6 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56626 A1	
	BCM56626 B0	
	BCM56626 B1	
	BCM56626 B2	
	BCM56628 A0	8 port 10-GbE/HiGig2 Multilayer Ethernet Switch with External Table Expansion
	BCM56628 A1	
	BCM56628 B0	
	BCM56628 B1	
	BCM56628 B2	
	BCM56629 B0	25 port 1-GbE Multilayer Ethernet Switch with 8 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56629 B1	
BCM56630	BCM56630 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56630 B0	
	BCM56634 A0	48-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56634 B0	
	BCM56636 A0	24-Port GbE + 2-Port 10-GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56636 B0	
	BCM56638 A0	4-Port 10-GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56638 B0	
	BCM56639 A0	24-Port GbE + 4-Port 10-GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56639 B0	
BCM56540	BCM56540 A0	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
	BCM56540 A1	
BCM56540	BCM56541 A0	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
	BCM56541 A1	
BCM56540	BCM56542 A0	28xGE + 2xF.XAUI/2x10GE + 2xF.HG[42] + 2xF.HG[21] + 1GE, 28xGE + 8xGE/8x2.5GE + 2xHG[42] + 2xHG[21] + 1GE Multilayer Ethernet Switch (Preview)
	BCM56542 A1	
BCM56540	BCM56544 A0	10xF.XAUI + 4xHG[21] + 1GE, 10xF.XAUI + 4xXFI, 10xF.XAUI + 2xHG[42], 4xXAUI + 12xXFI + 1GE Multilayer Ethernet Switch (Preview)
	BCM56544 A1	

Table 104: Switch Devices that support Warm boot

Family	Devices	Description
BCM56540	BCM56545 A0	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
	BCM56545 A1	
BCM56540	BCM56546 A0	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
BCM56640	BCM56640 A0	1x100GE + 1xHG[127], 1x100GE + 4xHG[32], 1x100GE + 8xHGd[16], 3xF.HG[42] + 1xHG[127], 3xF.HG[42] + 4xHG[32], 3xF.HG[42] + 8xHGd[16], 3xF.HG[42] + 3xF.HG[42] Multilayer Ethernet Switch (Preview)
	BCM56640 A1	
BCM56640	BCM56643 A0	48xGE + 4xXFI + 4xHG[42] + 1GE Multilayer Ethernet Switch (Preview)
	BCM56643 A1	
BCM56640	BCM56644 A0	48xGE + 2xHG[25] + 2xHG[25] + 1GE Multilayer Ethernet Switch (Preview)
	BCM56644 A1	
BCM56640	BCM56648 A0	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
	BCM56648 A1	
BCM56640	BCM56649 A0	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
BCM56680	BCM56680 A0	25 port 1-GbE/2.5GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports
	BCM56680 A1	
	BCM56680 B0	
	BCM56680 B1	
	BCM56684 A0	24 port 1-GbE/2.5GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports
	BCM56684 A1	
	BCM56684 B0	
	BCM56684 B1	
BCM56685	BCM56685 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56685 B0	
	BCM56689 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56689 B0	
BCM56700	BCM56700 A0	16-Port, 192-Gbps Lossless Switch Fabric
	BCM56701 A0	12-Port, 144-Gbps Lossless Switch Fabric
BCM56720	BCM56720 A0	16 Port, 16-Gbps HiGig2 Switch Fabric
	BCM56721 A0	12 Port, 16-Gbps HiGig2 Switch Fabric
BCM56725	BCM56725 A0	8 Port, 20-Gbps + 4 Port, 16-Gbps HiGig2 Switch Fabric
BCM56800	BCM56800 A0	20-Port 10-Gigabit Ethernet Multilayer Switch
	BCM56801 A0	10-Port 10-Gigabit Ethernet and 8-Port HiGig2/10GbE Multilayer Switch
	BCM56802 A0	16-Port 10-GbE/HiGig2 Multilayer Switch
	BCM56803 A0	12 Port 10GE/HiGig2 Multilayer Switch
BCM56820	BCM56820 A0	24 x 10-GbE + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56820 B0	
	BCM56821 A0	12 x 10-GbE + 8 x HiGig2 + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56821 B0	
	BCM56822 A0	12 x 10-GbE + 4 x 20-Gbps HiGig2 + 4 x 1-GbE Multilayer Ethernet Switch

Table 104: Switch Devices that support Warm boot

Family	Devices	Description
	BCM56822 B0	
	BCM56823 A0	8 x 10-GbE + 4 x 20-Gbps HiGig2 + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56823 B0	
	BCM56825 B0	16 x 10-GbE + 8 x 20-Gbps HiGig2 + 1 x 1-GbE Multilayer Ethernet Switch
BCM56840	BCM56841 A0	320 Gbps Ethernet Multilayer Switch
	BCM56841 A1	
	BCM56841 A2	
	BCM56841 A3	
	BCM56841 A4	
	BCM56841 B0	
	BCM56841 B1	
	BCM56843 A0	480 Gbps Ethernet Multilayer Switch
	BCM56843 A1	
	BCM56843 A2	
	BCM56843 A3	
	BCM56843 A4	
	BCM56843 B0	
	BCM56843 B1	
	BCM56845 A0	640 Gbps Ethernet Multilayer Switch
	BCM56845 A1	
	BCM56845 A2	
	BCM56845 A3	
	BCM56845 A4	
	BCM56845 B0	
	BCM56845 B1	
BCM56840_PLUS	BCM56842 A0	320 Gbps Ethernet Multilayer Switch
	BCM56842 A1	
	BCM56844 A0	480 Gbps Ethernet Multilayer Switch
	BCM56844 A1	
	BCM56846 A0	640 Gbps Ethernet Multilayer Switch
	BCM56846 A1	
BCM56850	BCM56850 A0	1.28Tbps I/O, 1Tbps Core Ethernet Switch
	BCM56854 A0	1.28Tbps I/O, 1Tbps Core Ethernet Switch
	BCM56850 A1	1.28Tbps I/O, 1Tbps Core Ethernet Switch
	BCM56854 A1	1.28Tbps I/O, 1Tbps Core Ethernet Switch
BCM88640	BCM88640 A0	80GBps DNX Traffic manager + Packet processor
	BCM88640 B0	
BCM88650	BCM88650 A0	200GBps DNX Traffic manager + Packet processor
	BCM88650 B0	
	BCM88650 B1	
BCM88660	BCM88660 A0	200GBps DNX Traffic manager + Packet processor
BCM88750	BCM88750 A0	1600GBps DNX Switch fabric

Table 104: Switch Devices that support Warm boot

Family	Devices	Description
	BCM88750 B0	

Note: There is no warm boot support for External table expansion in BCM56620, BCM56630 and BCM56640 device family.

PHYS

Table 105: PHYs

Device	*Driver Family*	*Description*
BCM5218	522x	10/100Base-TX/FX Octal-PHY(tm) Transceiver
BCM5220	522x	10/100BASE-TX/FX Mini-F(tm) Transceiver
BCM5221	522x	10/100BASE-TX/FX Mini-F(tm) Transceiver
BCM5226	522x	10/100 BASE- TX/FX Hex-PHY(tm) Transceiver
BCM5228	522x	10/100BASE-TX/FX Octal-F(tm) Transceiver
BCM5238	522x	10/100BASE-TX OCTAL-f(tm) Transceiver
BCM5248	522x	10/100BASE-TX Octal-F(tm) Transceiver
BCM52681E A1	54680	Octal 10/100 Ethernet Transceiver
BCM5401	5401	10/100/1000BASE-T Gigabit Copper Transceiver
BCM5402	5402	10/100/1000BASE-T Gigabit Copper Transceiver
BCM5404	5404	Quad-Port 10/100/1000BASE-T Gigabit Copper Transceiver
BCM5424	5424	Quad 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM5434	5424	Quad 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM5411	5411	10/100/1000BASE-T Gigabit Copper Transceiver
BCM5421	5421S	10/100/1000BASE-T Gigabit Copper Transceiver
BCM5421S	5421S	10/100/1000BASE-T Gigabit Copper Transceiver with SerDes
BCM5461	5464	10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM5464	5464	Quad-Port 10/100/1000BASE-T Gigabit Copper Transceiver
BCM5464R	5464	Quad-Port 10/100/1000BASE-T Gigabit Copper Transceiver
BCM5464S	5464	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM5464SR	5464	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM5466	5464	Quad-Port 10/100/1000BASE-T Gigabit Copper Transceiver
BCM5466R	5464	Quad-Port 10/100/1000BASE-T Gigabit Copper Transceiver
BCM5466S	5464	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM5466SR	5464	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM5482	5482	Dual-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM5488	5464	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54240_C0	54280	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54240_C1	54280	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54280_A0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54280_C0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54280_C1	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54282_A0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54282_C0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54282_C1	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54285_C0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54285_C1	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54290_A0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (1588 feature is Bring-up)
BCM54292_A0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (1588 feature is Bring-up)
BCM54294_A0	54280	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (1588 feature is Bring-up)
BCM54340_B0	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)

Table 105: PHYs

Device	Driver Family	Description
BCM54340_C0	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54340_C1	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54380_B0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54380_C0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54380_C1	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54382_B0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54382_C0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54382_C1	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54385_B0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54385_C0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54385_C1	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54616_A0	54616	Single-Chip 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54640	54640	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM54640E_A1	54640	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM54640E B0	54640	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM54680_A0	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54680E_A1	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54680E_B0	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54682E_A1	54682	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with 2 Copper/Fiber Media Interface
BCM54682E_B0	54682	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with 2 Copper/Fiber Media Interface
BCM54684_D0	54684	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54684E B0	54682	10/100/1000 Octal (65nm) QSGMII-Copper/Fiber(2) with EEE
BCM54685	54682	Octal QSGMII to 10/100/1000BaseT or Fiber Ethernet Transceiver
BCM54685E_A1	54682	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with Copper/Fiber Media Interface
BCM54810_A0	54880	BroadR-Reach Single-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54880_A0	54880	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with BroadR-Reach support
BCM54880_B0	54880	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with BroadR-Reach support
BCM54880E_A1	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54880E_B0	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54881_B0	54880	Octal 10/100Base/Tx Ethernet BroadReach Transceiver
BCM54942 A0	84728	Quad-Channel 10GbE XAUI-to-XFI PHY. Firmware version 0124
BCM54980_B2	54980	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54980_C0	54980	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54980_C1	54980	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM8040_A2	8040	Eight-Channel Multirate 1-Gbps - 3.2-Gbps Retimer/Switch
BCM8073_A0	8072	Dual-Channel Serial 10-GbE BASE-KR to XAUI Transceiver. Firmware version d502.

Table 105: PHYs

Device	Driver Family	Description
BCM8074_A0	8072	Quad-Channel Serial 10-GbE BASE-KR to XAUI Transceiver. Firmware version 010C.
BCM8704	8703	Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with XAUI Interface
BCM8705	8705	Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with WIS Layer and XAUI Interface
BCM8725	8705	Dual Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with WIS Layer and XAUI Interface
BCM8726_A0	8706	Dual Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with XAUI Interface
BCM8726_B1	8706	Dual Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with XAUI(TM) Interface. Firmware version 0x0127
BCM8727_B0	8706	Dual Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with XAUI Interface. Firmware version 0406.
BCM8727_C0	8706	Dual Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with XAUI Interface. Firmware version 050D.
BCM84727_A0	84728	Dual SFI to XAUI with 1588 (Firmware version 0x124. Bring-up)
BCM8728_A0	8706	Dual-Channel 10-GbE SFI-to-XAUI(TM) Transceiver with EDC. Firmware version 0511. (Bring-up)
BCM8742	8706	Quad-Channel 10-GbE SFI-to-XAUI(TM) Transceiver. Firmware version 0511.
BCM8747_A0	8706	Quad-Channel 10-GbE SFI-to-XAUI(TM) Transceiver with EDC. Firmware version 0511.
BCM8750_A0	8750	Dual-Channel 10 GbE SFI-to-XFI PHY with EDC
BCM8752_A0	8750	Dual-Channel 10 GbE SFI-to-XFI PHY with EDC
BCM8754_A0	8750	Quad-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version 0411.
BCM8481_B0	8481	10GBASE-T Transceiver (Firmware version B0 02.10)
BCM8481_C0	8481	10GBASE-T Transceiver (Firmware version C0 02.13)
BCM84164	BCM84740	Quad 10GBASE-KR-to-XFI or 40GBASE-KR4-to-XLAUI Transceiver Firmware version 0x128
BCM84168	BCM84740	Octal 10GBASE-KR-to-XFI or Dual 40GBASE-KR4-to-XLAUI Transceiver Firmware version 0x128
BCM84318_A0	84740	10.3 Gbps Octal Port CDR/Retimer with EDC. Firmware version D007
BCM82328_A0	82328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version 9 "(Bring-up)
BCM82328_B0	82328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version D "(Bring-up)
BCM84328_A0	84328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version D026
BCM84328_B0	84328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version D026
BCM84333_B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.69 (Bring-up) (Needs additional software component)
BCM84334_B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.69 (Bring-up) (Needs additional software component)
BCM84336_B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.69 (Bring-up) (Needs additional software component)
BCM84793_A0	84793	100GbE/OTN 4x25/28G VSR28 to 10x10/11G CAUI Gearbox PHY. Firmware version 0xD009 (Bring-up - Mode-1 and Mode-3)
BCM84812_A0	8481	Dual 10GBASE-T Transceiver. Firmware version 2.13
BCM84821_A0	8481	10GBASE-T Transceiver. Firmware version 2.13 (Bring-up)
BCM84822_A0	8481	Dual 10GBASE-T Transceiver. Firmware version 3.02
BCM84823_A0	8481	Dual 10GBASE-T Transceiver. Firmware version 3.02
BCM84823_B0	8481	Dual 10GBASE-T Transceiver. Firmware version 4.02
BCM84823_B1	8481	Dual 10GBASE-T Transceiver. Firmware version 4.02
BCM84833_B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.69(Driver support for IEEE 1588 features are Bring-up)

Table 105: PHYs

Device	Driver Family	Description
BCM84834_B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.69(Driver support for IEEE 1588 features are Bring-up)
BCM84836_B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.69(Driver support for IEEE 1588 features are Bring-up)
BCM84844_A0	8481	Quad 10GBASE-T Transceiver. Firmware version 1.07.11(Driver support is Bring-up)
BCM84846_A0	8481	Dual 10GBASE-T Transceiver. Firmware version 1.07.11(Driver support is Bring-up)
BCM84848_A0	8481	Quad 10GBASE-T Transceiver. Firmware version 1.07.11(Driver support is Bring-up)
BCM84728 A0	84728	Dual-Channel 10 GbE SFI-to-XAUI LAN/WAN PHY with 1588. Firmware version 0124 (Driver support for IEEE 1588 features is Bring-up)
BCM84729_A0	84729	Dual-Channel SFI to XAUI with Macsec, 1588 (Firmware version 0x124. Driver support for IEEE 1588 features are Bring-up)
BCM84740 A0	84740	40 GbE PPI-to-XLAUI PHY with EDC. Firmware version D106.
BCM84741 B0	84756	40GbE XLPPI-to-XLAUI/Quad 10G with IEEE MACsec/1588 Firmware version 0x0128 [Bring-up]
BCM84747_A0	84728	Quad SFI to XAUI with 1588 (Firmware version 0x124. Bring-up)
BCM84748_A0	84728	Quad SFI to XAUI with WAN/1588 (Firmware version 0x124. Bring-up)
BCM84749_A0	84749	Quad SFI to XAUI with Macsec, 1588 (Firmware version 0x124. Driver support for IEEE 1588 features are Bring-up)
BCM84752 A0	84740	Dual-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version D105. (Bring-up)
BCM84753 A0	84740	Quad-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version D105.
BCM84754 A0	84740	Quad-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version D105.
BCM84756 A0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version D105. (Needs additional software component)
BCM84756 B0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version 0x0128(Needs additional software component)
BCM84756 C0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version 0x0128(Needs additional software component) [Bring-up]
BCM84758	84740	10GbE Quad SFI-XFI PHY with IEEE 1588 Firmware version 0x128
BCM84759 A0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version D105.
BCM84759 C0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version 0x0128. (Bring-up)
BCM84780_A0	84740	Octal-Channel 10 GbE SFI-to-XFI PHY with 1588. Firmware version 0x128 (Bring-up)
BCM84784_A0	84740	Dual 40GbE/Octal 10GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version 0x125 (Bring-up)
BCM84764_A0	84728	Quad SFI to RXAUI with 1588 (Firmware version 0x124. Bring-up)
BCM84064 A0	84740	Quad 10G-KR-to-XFI or 40G-KR4-to-XLAUI Transceiver. Firmware version 0108.
BCM84074_A0	84728	Quad KR to XAUI (Firmware version 0x124. Bring-up)

OPERATING SYSTEMS

The SDK provides the SAL and BDE abstraction implementations necessary for running the SDK on the following operating systems. See the Platform Guide (56XX-PG818-R) for instructions on porting the SDK to another platform.

Table 106: Operating Systems

Operating System
VxWorks 5.5
VxWorks 6.2
VxWorks 6.4
VxWorks 6.5
VxWorks 6.6
Linux 2.6.21 User Mode
Linux 2.6.21 Kernel Resident Mode
Linux 2.6.25 User Mode
Linux 2.6.25 Kernel Resident Mode
Linux 2.6.27 User Mode
Linux 2.6.27 Kernel Resident Mode
Linux 2.6.35 User Mode
Linux 2.6.35 Kernel Resident Mode
POSIX Compliant (SAL ONLY)

CPU SUBSYSTEMS

Table 107: CPU Subsystems

CPU Subsystem	Description
BCM98245	CPCI 32-bit PPC with Motorola 8245 Processor
BCM98548XMC	XMC 32-bit PPC with Freescale 8548 Processor
BCM953003C	XMC 32-bit MIPS74Kc with BCM53003 Processor
BCM5300X	32-bit MIPS74Kc with BCM5300X Processor
BCM5301X	Integrated ARM Cortex-A9 CPU on BCM5301X Switch Devices
BCM5302X	Integrated ARM Cortex-A9 CPU on BCM5302X Switch Devices
BCM5621X	Integrated MIPS CPU on BCM5621X Switch Devices
BCM5622X	Integrated MIPS CPU on BCM5622X Switch Devices
BCM5331X	Integrated MIPS CPU on BCM5331X Switch Devices
BCM5360X	Integrated MIPS74Kc CPU on BCM5360X Switch Devices
BCM9XLP316LXMC	XMC with Broadcom XLP 316 processor that includes up to sixteen NXCPUs(4 cores)
BCM958625XMC(CPU)	XMC with BCM58625 processor (1.2 GHz ARMCortex.-A9 dual-core processor NXCPUs(4 cores) each operating at up to 2.0 GHz
BCM9XLP208XMC	XMC with Broadcom XLP 208 processor that includes up to eight NXCPUs(2 cores)

CPU AND OPERATING SYSTEM COMBINATIONS

The following CPU and Operating System combinations are supported by the SDK (in addition to the above):

Table 108: CPU and Operating System Combinations

CPU Subsystem	Operating System	Description
BCM98245	VxWorks 6.2	BSP Provided
BCM98245	Linux 2.6.21	Available through WindRiver Linux 2.0
BCM5621X	VxWorks 6.4	BSP Provided
BCM5621X	Linux 2.6.21	Available through WindRiver Linux 2.0 `bcm_ntswics`
BCM5331X	VxWorks 6.4	BSP Provided
BCM5331X	Linux 2.6.21	Available through WindRiver Linux 2.0 `bcm_ntswics`
BCM98548XMC	VxWorks 6.5	BSP Provided
BCM98548XMC	Linux 2.6.27	Available through WindRiver Linux 3.0. Note: Additional patches for issues WIND00172598 and WIND00161649 are required. Contact your WindRiver support personnel for these patches and other WindRiver information.
BCM5300X	VxWorks 6.6	BSP Provided
BCM5300X	Linux 2.6.21	Available through WindRiver Linux 2.0
BCM5300X	Linux 2.6.27	Available through WindRiver Linux 3.x
BCM5301X	Linux 2.6.35	Available through Broadcom Customer Support Portal
BCM5302X	Linux 2.6.35	Available through Broadcom Customer Support Portal
BCM5360X	VxWorks 6.6	BSP Provided
BCM5360X	Linux 2.6.21	Available through WindRiver Linux 2.0
BCM5360X	Linux 2.6.27	Available through WindRiver Linux 3.x
Generic X86	Linux 2.6.25/2.6.27	

Section 8: Release Media

The Software Development Kit is released as a gzipped tar file on the Broadcom Customer Support Portal, http://support.broadcom.com. The Network Switching Software Platform Guide, also available on the Customer Support Portal, provides documentation on the various components, the source directory layout, how to build the release for various platforms, and how to customize and port the software to new platforms.

Section 9: Support

Questions, feedback, and/or suggestions should be sent to your Broadcom FAE.

Section 10: Firmware Compatibility Matrix

The following table shows compatibility between different versions of SDK and Firmware releases.

BCM56440 FIRMWARE COMPATIBILITY MATRIX

Table 109:

SDK	Firmware 3.0.0	Firmware 3.0.1	Firmware 3.1.0	Firmware 3.2.0	Firmware 3.2.1	Firmware 3.2.2	Firmware 4.0.0	Firmware 4.0.1	Firmware 4.0.2	Firmware 4.0.3
SDK-6.2.9	Yes	Yes	Yes	No	No	No	No	No	No	No
SDK-6.3.0	Yes	Yes	No	Yes	No	No	No	No	No	No
SDK-6.3.1	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.2	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.3	No	No	No	Yes*	Yes*	Yes*	No	No	No	No
SDK-6.3.4	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.5	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.6	No	No	No	Yes	Yes	Yes	Yes	No	No	No
SDK-6.3.7	No	No	No	Yes	Yes	Yes	Yes	Yes	No	No
SDK-6.3.8	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	In validation
SDK-6.3.9	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SDK-6.4.0	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.4.1	No	No	No	No	No	No	No	No	Yes	In validation
SDK-6.4.2	No	No	No	No	No	No	No	No	Yes	In validation

* In order to support embedded applications, a patch is needed to merge the fix for SDK-53008 to SDK 6.3.3 release.

BCM56640 FIRMWARE COMPATIBILITY MATRIX

Table 110:

SDK	Firmware 3.0.0	Firmware 3.0.1	Firmware 3.1.0	Firmware 3.2.0	Firmware 3.2.1	Firmware 3.2.2	Firmware 4.0.0	Firmware 4.0.1	Firmware 4.0.2	Firmware 4.0.3
SDK-6.2.9	No	No	Yes	No	No	No	No	No	No	No
SDK-6.3.0	Yes	Yes	No	No	No	No	No	No	No	No
SDK-6.3.1	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.2	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.3	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.4	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.5	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.6	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.7	No	No	No	Yes	Yes	Yes	Yes	Yes	No	No
SDK-6.3.8	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	In validation
SDK-6.3.9	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SDK-6.4.0	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.4.1	No	No	No	No	No	No	No	No	Yes	In validation
SDK-6.4.2	No	No	No	No	No	No	No	No	Yes	In validation

BCM88650 FIRMWARE COMPATIBILITY MATRIX

Table 111:

SDK	***Firmware 3.0.0***	***Firmware 3.0.1***	***Firmware 3.1.0***	***Firmware 3.2.0***	***Firmware 3.2.1***	***Firmware 3.2.2***	***Firmware 4.0.0***	***Firmware 4.0.1***	***Firmware 4.0.2***	***Firmware 4.0.3***
SDK-6.2.9	No	No	Yes	No	No	No	No	No	No	No
SDK-6.3.0	Yes	Yes	No	Yes	No	No	No	No	No	No
SDK-6.3.1	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.2	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.3	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.4	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.5	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.6	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.7	No	No	No	Yes	Yes	Yes	Yes	Yes	No	No
SDK-6.3.8	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	In validation
SDK-6.3.9	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SDK-6.4.0	No	No	No	Yes	Yes	Yes	No	No	No	No
SDK-6.4.1	No	No	No	No	No	No	No	No	Yes	In validation
SDK-6.4.2	No	No	No	No	No	No	No	No	Yes	In validation

BCM56850 FIRMWARE COMPATIBILITY MATRIX

Table 112:

SDK	*Firmware 3.1.0*	*Firmware 3.2.0*	*Firmware 3.2.1*	*Firmware 3.2.2*	*Firmware 4.0.0*	*Firmware 4.0.1*	*Firmware 4.0.2*	*Firmware 4.0.3*
SDK-6.2.9	No	Yes	No	No	No	No	No	No
SDK-6.3.0	No	Yes	No	No	No	No	No	No
SDK-6.3.1	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.2	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.3	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.4	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.5	No	Yes	Yes	Yes	No	No	No	No
SDK-6.3.6	No	Yes	Yes	Yes	Yes	No	No	No
SDK-6.3.7	No	Yes	Yes	Yes	Yes	Yes	No	No
SDK-6.3.8	No	Yes	Yes	Yes	Yes	Yes	Yes	In validation
SDK-6.3.9	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SDK-6.4.0	No	Yes	Yes	Yes	No	No	No	No
SDK-6.4.1	No	No	No	No	No	No	Yes	In validation
SDK-6.4.1	No	No	No	No	No	No	Yes	In validation

BCM88030 FIRMWARE COMPATIBILITY MATRIX

Table 113:

SDK	*Firmware 3.2.0*	*Firmware 3.2.1*	*Firmware 3.2.2*	*Firmware 4.0.0*	*Firmware 4.0.1*	*Firmware 4.0.2*	*Firmware 4.0.3*
SDK-6.2.9	Yes	No	No	No	No	No	No
SDK-6.3.1	Yes	Yes	Yes	No	No	No	No
SDK-6.3.2	Yes	Yes	Yes	No	No	No	No
SDK-6.3.3	Yes	Yes	Yes	No	No	No	No
SDK-6.3.4	Yes	Yes	Yes	No	No	No	No
SDK-6.3.5	Yes	Yes	Yes	No	No	No	No
SDK-6.3.6	Yes	Yes	Yes	Yes	No	No	No
SDK-6.3.7	Yes	Yes	Yes	Yes	Yes	No	No
SDK-6.4.0	Yes	Yes	Yes	No	No	No	No
SDK-6.4.1	No	No	No	No	No	Yes	In validation
SDK-6.4.2	No	No	No	No	No	Yes	In validation

BCM56450 FIRMWARE COMPATIBILITY MATRIX

Table 114:

SDK	*Firmware 3.0.0*	*Firmware 3.0.1*	*Firmware 3.1.0*	*Firmware 3.2.0*	*Firmware 3.2.1*	*Firmware 3.2.2*	*Firmware 4.0.0*	*Firmware 4.0.1*	*Firmware 4.0.2*	*Firmware 4.0.3*
SDK-6.3.6	No	No	No	No	No	No	Yes	No	No	No
SDK-6.3.7	No	No	No	No	No	No	Yes	Yes	No	No
SDK-6.3.8	No	No	No	No	No	No	Yes	Yes	Yes	In validation
SDK-6.3.9	No	No	No	No	No	No	Yes	Yes	Yes	Yes
SDK-6.4.0	No	No	No	No	No	No	No	No	No	No
SDK-6.4.1	No	No	No	No	No	No	No	No	Yes	In validation
SDK-6.4.2	No	No	No	No	No	No	No	No	Yes	In validation

BMACSEC SDK COMPATIBILITY MATRIX

Table 115:

Switch SDK Release	*BMACSEC SDK Release*
5.10.2	3.1
5.10.3	3.2
6.0.1	3.3
5.10.4	3.4
6.0.2	3.4
6.2.0	3.5
5.11.0	3.6
6.2.1	3.7
6.2.2	3.8
6.2.3	3.8
5.11.1	3.9
6.2.4	3.9
6.2.5	3.10
6.2.6	3.11
6.2.7	3.12
6.2.8	3.13
6.2.9	3.14
6.3.0	4.0
6.3.1	4.1
6.3.2	4.2
6.3.3	4.3
6.3.4	4.4
6.4.0	4.5
6.3.5	4.6
6.3.6	4.7
6.3.7	4.8
6.4.1	4.8
6.3.8	4.9
6.3.9	4.10
6.4.2	4.10

Section 11: SDK Externally Licensed Software Components

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.

Table 116: EXTERNALLY LICENSED SOFTWARE COMPONENTS

Component	*Origin*	*Location in source tree*	*License terms and conditions*
EDITLINE	/afs/athena.mit.edu/contrib/sipb/src/editline	src/sal/appl/editline	See (EDITLINE License terms and conditions) (page 216)
ED Editor	USENET comp.sources.misc Volume 9, Issue 36	src/appl/diag/edline.c	See (ED Editor License terms and conditions) (page 218)
CINT	http://www.gnu.org/software/bison/	src/appl/cint/cint_parser.[ch]	See (CINT parser license terms and conditions) (page 219)
CES Driver	BATM Advanced Communications Ltd	src/soc/ces/nemo_driver/*.[ch], src/soc/ces/clsbuilder/*.[ch]	See (Circuit Emulation Service (CES) Driver terms and conditions) (page 220)
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	See (BIGDIGITS license terms and conditions) (page 221)
APIMODE	http://www.gnu.org/software/bison/	src/appl/diag/api/api_grammar.tab.[ch]	See (APIMODE parser license terms and conditions) (page 222)
VxWorks	Wind River Systems, Inc.	systems/vxworks	See (Wind River Systems license terms and conditions) (page 223)

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.

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 'ˆ' 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
```

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.  */
```

CIRCUIT EMULATION SERVICE (CES) DRIVER TERMS AND CONDITIONS

The Circuit Emulation Services (CES) driver code provided herewith is provided by BATM Advanced Communications Ltd (BATM) and is subject to licensing agreement between BATM and Broadcom Corporation.

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.

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 APIMDOE to the FEATURE_LIST in SDK compilation flags.

See (CINT parser license terms and conditions) (page 219) for the Bison licence.

WIND RIVER SYSTEMS LICENSE TERMS AND CONDITIONS

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

Section 12: Resolved Issues for 6.4.1

The following issues are resolved in version 6.4.1 of the SDK.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-32461		56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56746_A0 56745_A0 56744_A0 56743_A0	Problem: WRED thresholds were not taking effect because of hardware issue. Solution: Implemented workaround in software to get WRED memories into stable state. This workaround does below thinks to put WRED memories in stable state. 1. Selects 4 Ethernet ports (one extended queue port from X,Y pipeline and one regular port from X, Y pipeline. 2. Configures all 4 ports in MAC loopback. 3. Disables CRC re-calculation on all 4 egress ports. 4. Enables CRC checks and configures ING_PRI_CNG_MAP table to mark incoming traffic with red color. 5. Add's l2 mac address in l2 table to switch the packets to all 4 ports. 6. Prepares 8 multi cell unicast SOBMH packets, configured DMA descriptor's and starts DMA engine. These SOBMH packets will be loopbacked with bad CRC since we disabled CRC re-calculation on egress ports and switches to all 4 egress ports. Since these packets have CRC errors MMU will drop the packets after receiving EOP and stabilizes the WRED memories.
SDK-34523		56820_A0 56820_B0	In previous SDK, customer found a crash on 56820 when the SDK was handling a MMU parity error. The root cause of this crash was the SDK visited an unavailable memory. Now this issue has been resolved.
SDK-36232	460304	All 56850_A0	In previous release, A L2 multicast with flag BCM_MULTICAST_WITH_ID and Group_ID was created by bcm_multicast_create, but the HW index in Group_ID was already occupied by other multicast group, than the existing entry could be overwrote and return BCM_E_NONE. In this release, it will return BCM_E_EXISTS and won't overwrite the existing entry.
SDK-38881		All	bcm_port_priority_color_set is modified to set color as none when color param to API is bcmColorPreserve.
SDK-41357	469082	56842_A0	There is an issue with the h/w logic related to the parity generation and checking for the PORT_CBL_TABLE memory. In this release occasional spurious reports of a parity error in PORT_CBL_TABLE has been fixed.

Table 117:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
SDK-42031		`88650_A0`	Error indication was added to prevent AC P2P to PWE or to Mac-In-Mac inner vlan editing (VID-2 in LIF table) which is not supported by HW.
SDK-42289	565794	`88650_A0`	Static forwarding (i.e. `bcm_port_force_forward_set` API) can be used both in TM and PP modes. Some fixes are done to enable it also in TM mode.
SDK-42527		`88650_A0`	SDK-42527: Support TR 90 and TR91 for the ARAD.
SDK-42957	580600	`88025_A0`	Support for Down MEP on VPWS/VPLS attachment circuits is fixed.
SDK-42987	580192	`56850_A0`	Legacy method to add route entries to LPM table may trigger re-shuffling logic which could lead to massive HW entry movement. In the worst case, the memory read operation times of LPM tables could be very high and eventually cause bad route convergence time. A request was received to reduce the totaled time costs for bulk route add operation. This was achieved by enabling soc memory cache for route add/delete operation to reduce memory read time costs. Can be turned on/off by switch control bcmSwitchL3RouteCache. Read HIT* bits could be wrong during caching time.
SDK-44506	593957	`56842_A0`	Added a new soc property `(L3_DISABLE_ADD_TO_ARL)` to restrict applications to create L2 interface entry and L3 interface entry separately. SDK uses this property to recover the association between L2 entries and L3 entries during warmboot. Currently, during warmboot, SDK associates L3 entries with L2 entries assuming that they were created with `BCM_L3_ADD_TO_ARL` flag during `l3_intf_create.` Later, when L3 interface is deleted, SDK deletes the L2 entry also.
SDK-44591		`56840_A0 56640_A0 56640_A1 56640_B0`	Current implementation is not in-line with the issue. Function: `wcmod_esm_serdes_control_get(int` unit, `int` lane, `soc_phy_control_t` type, `uint32` *value) case `SOC_PHY_CONTROL_DUMP:` rv = `wcmod_uc_status_dump` (unit, port, `NULL);` break; 'value' variable is not being used in this call.
SDK-44736		`56850_A1`	The 3-lane TSC configuration is now transcribed properly to TSC 31.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-44989		88660_A0	Supporting OAMP protection packets in 88660. To enable this feature, call bcm_rx_trap_type_create() with the flag WITH_ID, trap_type bcmRxTrapOampProtection and a trap id in the range 0x400 0x4ff, followed by bcm_rx_trap_set() with the trap id created in the above API, and a bcm_rx_trap_config_t with the field dest_port set to the destination of the protection packets. All other fields should remain blank (an example of this is found in cint_oam.c). Whenever an OAM event occurs, a protection packet of size 71 bytes will be sent to the destination selected above. The format of the Protection packet at the CPU will be FTMHoPPHoFHEI. The FHEI.CPU-TRAP-CODE field will be set to the LSB of the trap id selected in bcm_rx_trap_type_create(). The size of the protection packet will always be 71 bytes where the OAM events will be on the bottommost part of the packet.
SDK-45246		56840_A0	Implemented "bcmFieldActionL3ChangeMacDa" and "bcmFieldActionL3ChangeVlan" actions for TD2 device,TR3 and KATANAx devices. The actions expect the egress-object (l3 next hop index) which should be already created by L3 module and the ID should not be associated with any other modules. The actions are to replace the destination MAC and VLAN on the matched incoming packet with the MAC and VLAN associated with the given next hop index.
SDK-45535		88650_B0	OAM endpoint: It is now possible to create an OAM endpoint over LAG. This is done by adding the endpoint on the LAG port and separately configuring the mac address of the endpoint on each one of the LAG ports using bcm_l2_station_add. An example can be found in cint_oam_over_endpoint.c In addition CINT includes a cleanup function, and an option to set VLAN-Ports lifs over lag without defining an OAM endpoint.
SDK-46635	625709	56640_A0 56640_A1 56640_B0	Added a new SOC property "ext_tcam_request_response_latency" and a new "tcam latency" bcm shell command for TR3 with external TCAM. Customers can execute the bcm command to calculate the latency and then specify the SOC property using the latency value in config file.
SDK-46641	633505	88650_A0 88650_B0	When running 802.3 llc packets, the ethertype field is used as length. There was a bug that if the length was set to 0, the packet would have been parsed as a trill packet.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-46757	636270	56643_A0	Triumph_3 has a Unified Forwarding Table and hash selection for L2/L3/Vlan/Mpls tables has to be programmed differently. HASH_CONTROL register does not have a L3_HASH_SELECT field. The command "l3 l3table hash" is trying to access this non-existent field resulting in an assertion failure. Support for "l3 l3table hash" and "l3 l3table ip6hash" commands have been implemented.
SDK-46833		56440_B0	The fix for this issue checks the PLL's current divider setting in register 0x8050, and then use this as the forced value instead of always forcing the PLL to the same frequency for PRBS test. Additionally, the asymmetric mode for the 40nm B0 core has been disabled for the PRBS function to work.
SDK-47665	650917	56854_A2 56854_B0 56854_A0	1G configs should be supported in latest release.
SDK-47824	636400	56846_A0 56845_B0 56845_A2 56844_A0 56842_A0	In previous releases, the updating on NONUCAST_TRUNK_BLOCK_MASK table took a fair amount of time to complete on Trident+ which has more NONUCAST_TRUNK_BLOCK_MASK table entries with the plain read API soc_read_mem(). In this release the performance has been improved by changing the update method such that it now uses a single DMA operation soc_read_mem_range() which gives a good amount of speedup to applicable trunk APIs.
SDK-47983	661534	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2	New cosq type bcmCosqControlEgressPortPoolYellowLimitBytes / bcmCosqControlEgressPortPoolRedLimitBytes have been added for configuring yellow/red limits. Added one service pool type bcmCosqServicePoolPortColorAware and bcm_cosq_service_pool_set/get APIs for per port per service pool color aware enable.
SDK-48016	661903	56840_A0	In the previous release,static multicast L2 entries were getting flushed with bcm_l2_addr_delete_by_port() API.This has been resolved.
SDK-48101	689094	56845_B0 56845_A2 84740_A0 84784_A0	Support for 40G repeater mode PRBS for the system side was missing for BCM84740. Support for the same is added and tested in this release.
SDK-48140		88650_A0	TRILL BEHAVIOR CHANGE. According to trill fgl rfc, at ingress trill fgl, native outer and inner tpids must have value 0x893b. So far in Trill application, native Ethernet tpids set outer tpid = 0x8100 and native inner tpid = 0x893b. New implementation is now aligned to trill fgl rfc. At ingress trill fgl, both native inner and outer tpids have value 0x893b. Settings are done using VLAN-editing and work for both normal vlan translation and Advanced modes.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-48296		88650_A0	When working with external TCAM, a master-key is sent from BCM886XX to KBP device with all the necessary fields for the forwarding and external ACL lookups. A diagnostic has been built to show the order of the fields in the master-key and master-result: BCM> kbp print master
SDK-48404	654018	56845_B0	For BCM56845m `phy_wc40_ability_remote_get` was unable to get the correct ability when the link partner did not enable CL73. Corrected the ability to obtain remote ability when the link partner doesn't enable CL73 on TD+/WC40.
SDK-48577		56640_A0 56643_A0 56340_A0 56640_A1 56643_A1 56640_B0 56643_B0 56540_B0 56045_B0 56040_A0 56547_A0	Added cpu based UFT mem sweep to detect and fix parity errors. Fixed issues found with graceful lookup error handling.
SDK-48774 SDK-56539	672146	88650_A0 88650_B1 88660_A0	IMPORTANT CHANGE (MIRROR SEQUENCE): RSPAN Mirroring: ingress and egress settings have been decoupled. Mirroring into RSPAN is now done in the following way: 1. Create L3 interface (this has not been changed). 2. Set a `bcm_tunnel_initiator_t` object with `type=bcm_TunnelTypeRspan,` vlan, tpid, `pkt_pri` configured as desired and call `bcm_tunnel_initiator_create().` This allocates entries in the EEDB. 3. Set a `bcm_mirror_destination_t` with the flag `BCM_MIRROR_DEST_TUNNEL_WITH_ENCAP_ID` set and with the `encap_id` field set to the `tunnel_id` returned from `bcm_tunnel_initiator_create()` using the macro `BCM_GPORT_TUNNEL_ID_GET().` Getting information on the RSPAN tunnel may be done by setting a `bcm_l3_intf_t` object with the field `l3a_tunnel_idx` set to the `tunnel_id` returned from `bcm_tunnel_initiator_create(),` via the macro `BCM_GPORT_TUNNEL_ID_GET().` Destroying the tunnel (freeing the EEDB entries) may be done with the API `bcm_tunnel_initiator_clear(),` with the `l3a_tunnel_idx` field set as in the get() API. Destroying the mirror not been changed. For an example see `mirror_with_rspan_example()` in `cint_mirror_erspan.c`
SDK-49047		88650_B0 88650_B1 88660_A0	1588 packets were stamped while received/send from/to ports in which 1588 was disabled. This is fixed. ISSU perspective: The fix supports ISSU if all the ports supporting 1588 are disabled before ISSU, and enabled after ISSU.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-49202		56640_A0 56640_A1 56640_B0	SOC properties ext_tcam_tx_driver_current, ext_tcam_tx_postcursor_tap, ext_tcam_tx_main_tap have been added to describe driver_current, postcursor tap and main tap for NL11K serdes TX direction parameters, and SOC property ext_tcam_rx_gain has been added to describe RX gain for Serdes RX direction parameter.
SDK-49205		56640_A0 56640_A1 56640_B0	Support for ESM interrupt was added in CMIC level interrupt handler. Once the ESM fatal errors are detected, the new-implemented "esm recovery" thread will be woken to restore ESM.
SDK-49249		88650_A0 88650_B0 88650_B1 88660_A0	DPOE application: Added an application example to show how to classify L2/L3/L4 header field and mapping frames to PON LIFs. More information is provided in cint_pon_dml_fec_app.c.
SDK-49543	663298	88650_A0 88650_B0 88660_A0	Fixed ARAD ports Leds in Negev chassis (updated the Led microprocessor program to match recent changes in $SDK software)
SDK-49694		56640_B0 56850_A1 56850_A2	Please use lpm_scaling_enable=1 in config to have the ability to add 64bv6 entries in paired tcam. When this config is enabled, V4,64B V6 entries can be added in the unreserved paired tcam. If lpm_ipv6_128b_reserved=0, then no tcam space is reserved for 128B V6 entries and complete paired TCAM can be used for 128BV6, V4, and 64B V6 entries. Please note that each entry of 64B V6 entry in the paired TCAM uses 2 indexes of L3_DEFIP view where as in unpaired TCAM ,it uses only 1 entry
SDK-50216	693383	56850_A0	In previous release, per VLAN VP replication was automatically enabled when a Gport adds to the VLAN. In this release, support has been added to control VP replication by bcm_vlan_control_vlan_set.
SDK-50389	695476	2000_A1	QE2000 fix when updating QOS parameters for a given multicast queue. The unicast queue configuration was incorrectly being updated when egress independent flow control is enabled.
SDK-50431		88660_A0	ERSPAN on XGS MAC extender system is now supported

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-50591		`88650_A0 88650_B0 88660_A0`	TRILL: For TRILL UC and MC egress RBridges, trill packets are classified to inLIFs whose IDs are always 0 by PORT VLAN Domain X Outer VLAN. A problem occurs when doing same interface filter. The inLIF ID and outLIF ID of trill packets of UC and MC egress RBridges are all 0. Packets are all dropped even in case it shouldn't be. A new program is added to classified trill packets into valid inLIF IDs. It's enabled by adding a soc property `"custom_feature_trill_designated_vlan_inlif_<port>=<lif_id>".` Once feature is enabled, soc properties should be enabled for all TRILL ports.
SDK-50755		`88650_A0 88650_B0 88650_B1 88660_A0`	Diag: "diag cosq qpair e2e ps=x" can be used to display e2e port scheduler model. Improvement in 1) adding new diagnose "diag cosq qpair e2e ps=x" 2) using "diag cosq qpair egq" instead of "diag cosq egq"
SDK-50760		`88650_A0 88650_B0 88650_B1 88660_A0`	Diag: "diag cosq `print_flow_and_up"` can be used to display scheduler model with given voq. Improvement in 1) adding port scheduler level between HR level and Port level 2) using "voq connector" in the diag command instead of "flow"
SDK-50899	634474	`56845_B0`	Updated Documentation for WRED Flags
SDK-51038	683239	`88640_A0`	Petra-B 88640 Ingress mirroring : Ingress mirror can't mirror the original packet for the Petra hardware limitation, the workaround for it is to configure the mirror port as RAW in config.bcm as `tm_port_header_type_2.BCM88640` = RAW. See `cint_petra_mirror_tests.c` for more information.
SDK-51292	708102	`56640_A0`	The HG capable dynamic ports Indexing offset was not accounted for on the SC/QM queues which led to wrong indexing for the dynamic ports.This has been fixed in the offset.
SDK-51352	708790	`56846_A0 56840_A0 56846_A1`	In previous versions, the routine `_soc_trident_mem_parity_control()` returned directly after configuring parity control for X-pipe and left Y-pipe parity control register un-configured for dual pipe IPIPE/EPIPE memories. In this version, `_soc_trident_mem_parity_control()` has been modified to configure both X-pipe and Y-pipe parity control.
SDK-51353		`56643_B0`	The new support for the below port configuration has been implemented. Device =56643 Frequency (MHz)= 450 Option = 4 GbE Port Group (XC[12:0]) = 36 x GbE+1 x GbE High Speed Port Gr 1 (WC[2:0])= 4 x XFI High Speed Port Gr 2 (WC[6:3])= 2 x HG[42] + x F.H [42]" AXP Port Guaranteed Bandwidth = 5G

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-51360	692893	56840_A0	When any module in SDK uses new stat APIs, the running STAT version changes from LEGACY to NEW. Currently VLAN Field Processor implementation works only with old stat APIs for older devices where advanced flex counters are not available. Hence, required support is added in VLAN Field Processor implementation to use new stat APIs for older devices that switched to NEW stat version.
SDK-51380		56440_A0 56440_A1 56440_B0	Enabled proper debug prints when API bcm_policer_group_create() fails.
SDK-51392 SDK-51964	710405	All	The BSL improvement has fixed this issue.
SDK-51464		88650_A0 88750_B0 88660_A0 88750_A0 88650_B0 88650_B1	Source-routed data cells, generated by CPU, can be transmitted and received by Fabric Element (FE) and FAP devices (over fabric interface). These cells are routed according to the specific path information they carry, while disregarding the fabric reachability information. These messages are used mainly for debug and diagnostics purposes, but can be also used for CPU-to-CPU messaging. The previous SDK versions supported this features using SoC APIs. Instead, new BCM APIs created: typedef struct bcm_fabric_route_s{ uint32 pipe_id; /* Origin fabric pipe */ uint32 number_of_hops; /* corresponds to the number of routing hops (number traversed links) */ int* hop_ids; /* traversed links */ } bcm_fabric_route_t; int bcm_fabric_route_tx(int unit, uint32 flags, bcm_fabric_route_t * route, uint32 data_in_size, /* input payload size */ uint32 *data_in /* input payload buffer */); int bcm_fabric_route_rx(int unit, uint32 flags, uint32 data_out_max_size, /* maximal size of the payload buffer */ uint32 *data_out, /* output payload buffer */ uint32 *data_out_size /* actual output payload size */);
SDK-51494		88650_B0	Fix cint_mpls_lsr.c function mpls_add_php_entry. next protocol flag was overwritten by BCM_MPLS_SWITCH_TTL_DECREMENT flag.
SDK-51525	677768	88030_A0	There was a bug in the C3 model and the XML based test framework wherein any changes to the configuration files (files of the type g3p1_<xyz>_cfg.lrp) did not take effect until the second run of the tests. This is because the models read in the existing configuration files first and the assembler updated them later. As of this release of the MDE, this has been fixed.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-51570		56850_A0 56850_A1	In previous release, NIV VP class-id setting was not supported by bcm_port_class_set/get API. In this release, support was added for setting NIV VP class-id by bcm_port_class_set/get API.
SDK-51601	708490	88030_B0 88030_A0	Egress filter issue
SDK-51617	710438	56450_A0	Issue was happening due to incorrect buffer length calculation. Function _soc_mem_array_sbusdma_write() is modified to use chunk_entries to write buffer with correct length.
SDK-51625		88650_A0 88650_B0 88650_B1	For debug reasons, an HW register is used to store the SW version used at init, and during ISSU.
SDK-51648	713425	56340M_A0 56640_A0 56340_A0 56640_A1 56640_B0	Added in the support for different freq. QG_PLL and WC_PLL for chipsets which have the H/W capability.
SDK-51658		88650_A0 88650_B0 88650_B1 88660_A0	Support the following APIs to replace properties without replacing Out-LIF discard indication: 1. bcm_l3_egress_create 2. bcm_mirror_destination_tunnel_create. 3 bcm_mpls_tunnel_initiator_create. 4. bcm_tunnel_initiator_create.
SDK-51707	715469	All	Optimized the ipmc performance if change 32K ipmc group from one ipmc index to another.
SDK-51725		56624_B0	SER support has been added for the following memories as part of this fix: MMU_WRED_CFG_CELL MMU_WRED_THD_0_CELL MMU_WRED_THD_1_CELL MMU_WRED_CFG_PACKET MMU_WRED_THD_0_PACKET MMU_WRED_THD_1_PACKET MMU_WRED_PORT_CFG_CELL MMU_WRED_PORT_THD_0_CELL MMU_WRED_PORT_THD_1_CELL MMU_WRED_PORT_CFG_PACKET MMU_WRED_PORT_THD_0_PACKET MMU_WRED_PORT_THD_1_PACKET
SDK-51810		88650_B1 88660_A0	Fixed three errors related to bcm_vlan_port_find: 1. When calling the API on an unprotected port, the failover_port_id field will be 1 instead of 0. 2. Any information related to 1+1 protection (ingress_failover_id, failover_port_id) was not filled when calling the API. ingress_failvoer_id and failover_port_id will now be filled when calling the API. 3. Added missing validations to function parameters.
SDK-51828	716994	56440_A0 56850_A0	Added new API bcm_stat_flex_pool_info_multi_get to retrieve the usage of flex counters in a pool
SDK-51906	699612	56450_A0	Due to flex operation issue on Cfg#12 with TDM-A2, used TDM-A3. Also corrected total slots required for TDM-A3 and removed one warning wrong comment.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-51936		`56850_A2`	For cl36 PRBS bus width must be set to 80 bits instead of 66 bits.
SDK-51997		`88660_A0`	In BCM88660, in Field Processor, a new feature for field comparison is added. In HW, the comparison is performed on Key D in the second cycle of the PMF. It compares the two halves of the key (80 LSB bits and 80 MSB bits) and writes the result to the 5 MSB bits of the key (bits 159:155). Each bit of the result indicates a match of 20 bits, such that the 4 LSB bits correspond to 20 bits of the key, and the 5th MSB bit indicates match of the full key (80 bits). HW limitation: In order to act upon match the field group in the MSB must be Direct Extraction, and the 32 bit key is taken from the MSB bits (159:128). Direct extraction can filter up to 4 bits, thus only the 4 LSB bits of the compare result are used (it covers the full key). In SW, the sequence to enable the new compare feature is as follows: 1. Add a field group (max 80 bits each) with `bcm_field_group_config_create()` and set `BCM_FIELD_GROUP_CREATE_IS_EQUAL` flag in group. This field group will use 80 LSB bits of the key. 2. Add another field group (mode = Direct Extraction) using `bcm_field_group_config_create()` and set `BCM_FIELD_GROUP_CREATE_IS_EQUAL` flag in group. This Field Group must also add the qualifier bcmFieldQualifyIsEqualValue to its QSET. This field group will use 80 MSB bits of the key. * bcmFieldQualifyIsEqualValue qualifier is not properly part of the key (no HW instructions are allocated for it). It indicates that this key is written with the compare result. * The max size of this Field Group is 80 bits (although it is a Direct Extraction Field group, limited to 32 bits in general). The comparison is performed on the entire 80 bits and the Direct Extraction key is taken from the 32 MSB bits. * Note: the 5 MSB bits are overridden by compare result. The compare can be used in parallel to bcmFieldQualifyCascadedKeyValue qualifier. A new cint is added for example: `cint_field_dir_ext_compare_result.c`
SDK-52072	716983	`88660_A0`	ERSPAN: Fixing a bug in `bcm_tunnel_initiator_clear()`. When a ERSPAN tunnel is created through `bcm_tunnel_initiator_create()`, two EEDB entries were allocated but in `bcm_tunnel_initiator_clear()` only one was freed.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-52087	719039	56850_A0	A customer reported an issue with EPMC Egress_set performance. egress_set on 1000 IPMC index taking 20 seconds on 3 instances, That was happening because bcm_XXX_ipmc_egress_intf_set used the total interface number(max to 48K in some chips) to calculate the hash value for each port and cause the performance problem. In this release IPMC Egresss_set performance has been improved to take approximately half the time originally reported by reducing the total interface number to calculate and using accelerated method for MY_STATION_TCAM memory field access.
SDK-52242		56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0	HW works with a single granularity value for both CIR and EIR, once the granularity value is fixed(CIR), the maximum value for EIR becomes limited. Hence the issue. On high rates (EIR > CIR) the granularity value is fixed to EIR.
SDK-52246	720771	56450_A0	The subtag packet can be multicasted/ broadcasted to multiple CoE subports at a time from CPU by using switch logic (pkt->tx_pbmp along with flag pkt->flags=BCM_TX_ETHER). For this the user needs to create L2MC entry or a VLAN with the destination CoE subports as members. The packet would go through the IP-EP pipeline where the packet may be dropped by IP or EP depending on configuration. The suggestion is to send SOBMH packets from CPU, one by one to the CoE subports instead of using pkt->tx_pbmp.
SDK-52287	713097	88030_A0	Bit hash ID numbering now starts from 0.
SDK-52325 SDK-51797	721812	All	There is requirement from customer to perform a loopback test on a port, while making sure it looked like totally "down" from the outside, i.e. the link is down, no traffic leakage, etc. The modification of this JIRA is to add the support of MAC loopback on disabled port.
SDK-52339	722376	56850_A0 56850_A1 56850_A2	Two data error event flags were added. If a parity error is uncorrectable, the flag SOC_SWITCH_EVENT_DATA_ERROR_UNCORRECTABLE will be set when SDK reporting SOC_SWITCH_EVENT_PARITY_ERROR event to application. If a parity error is correctable, but the error correction fails, the flag SOC_SWITCH_EVENT_DATA_ERROR_FAILEDTOCORRECT will be set when SDK reporting SOC_SWITCH_EVENT_PARITY_ERROR event to application.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-52355		`56850_A0 56850_A1 56850_A2`	Support has been added for retry in mem insert and delete for hash tables. Inline hash memory recovery was implemented for insert and delete operations.When an insert/delete operation encounters a parity error, the inline recovery routine will be invoked. The inline recovery routine will calculate different hash buckets in different hash memory banks based on the entry that will be inserted/deleted, then restore the each bucket in these banks. For new-added hash key types in Trident2 hash tables, support for these key types in hash entry comparing routine has also been added.
SDK-52385	721101	`88030_B0 88030_A0`	Byte order changed in diags mem commands as requestd.
SDK-52386	722002	`88030_A0`	Retry 10 times if error happen, many location for `post_cmd/get_response` pair.
SDK-52389	721614	`56850_A0`	API has been added for populating egress etag qos mapping.
SDK-52412	678409	`56340_A0 84756_A0`	Issue: ------ When a 1G fiber SFP is installed into a BCM-84756 10G port with no fiber attached, the hardware linkscan declares the port link state as up. Rootcause: ---------- Hardware link scan can only probe a single bit in a single external phy or internal SerDes based link register on a per port basis. For PHYs such as 84756 which needs to probe more than 1 bit on more than 1 register to conclude per port based link up/down status, these PHYs must support a "squelch" function. When the squelching function is enabled/ configured, the PHY will bring system side link status up when the line side link status is up and vice versa. Then the hardware link scan can probe the internal SerDes to conclude link status. The squelching function is not supported in 84756 driver `(phy84756_fcmap.c)` yet. Fix: ---- Implemented Squelch function in `py84756_fcmap.c` which is the root cause for the issue mentioned in this JIRA. Tested the squelch function implementation with 1G, 10G ports for the system side and the line side link as follows. i) Enabled system side squelch, could observe system side link goes down when Tx is disabled on the line side. ii) Enabled line side squelch, could observe line side link goes down when Tx is disabled on the system side. Squelch function is not invoked from the init function of PHY84756 fcmap driver. So user/customer needs to call squelch function explicitly using `SOC_PHY_CONTROL_TX_LANE_SQUELCH` whenever they wish to enable squelch on either line side Or system side.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-52442	696223	56850_A2	In previous releases, an issue was reported in the parallel vertical scan operation, When attempting a 1-D slice vertical scan, the results would not come out if horizontal_min=0 and horizontal_max=0.In this release the TSC diagnostics interface has been modified to return the proper H right max and left max values.
SDK-52454		88650_A0 88660_A0	Basic bridging Egress multicast: We assume over VLAN APIs that egress multicast exist. Now no error is returned when egress multicast does not exist
SDK-52458		88650_B0 88660_A0	In L3 forwarding, when using external TCAM for forwarding tables, they were actually defined as ACL tables, resulting in a large software state and reduced efficiency in configuration. The forwarding tables in the external TCAM are now defined as LPM tables, reducing memory consumption and enhancing configuration periods. The API calling sequence remains identical.
SDK-52459		88660_A0	DEFAULT BEHAVIOR CHANGE (ARAD+ only). When using external TCAM for forwarding, RPF and forwarding searches were performed on duplicated databases. RPF and forwarding searches are now performed on a single database, using SIP and DIP respectively in search keys and resulting in increased (doubled) routing table capacity. Note that in case of IPv4/6 + RPF forwarding query, external ACL databases IDs are changed to 1 and 3. The actions sizes for ACL databases have also changed accordingly: The action size for ACL database 0 is 64 bits. The action size for ACL database 1 is 32 bits. The action size for ACL database 2 is 16 bits. The action size for ACL database 3 is 24 bits. All of the changes above apply only to ARAD+ devices. ARAD devices behavior remains unchanged.
SDK-52564		56850_A1	Fixed traffic drops observed with ingress-traffic after creation of L2GRE access port with match criteria as MATCH_PORT_VLAN.
SDK-52591	725728	56450_A0	Added support to enhance the number of child nodes per scheduler node in BCM56450. The restriction of maximum of 64 child nodes has been removed in cosq APIs.
SDK-52636		88030_A0	Added support for 4x10G_20x1G_1xHG TDM with specific assignment of Warp Core to CLPORT & XTPORT
SDK-52650		56960_A0	Added new Port Prbs Polynomial type BCM_PORT_PRBS_POLYNOMIAL_X58_X31_1.
SDK-52734		88650_A0 88650_B0 88660_A0	Indication if the Warmboot is supported by the device added to avoid Warmboot errors in regression for a version where Warmboot mode is not compiled.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-52751	726121	56545_A0 56545_A1 56545_B0	In the previous release bcm_l2_cache_set() did not allow setting priority > 7 on Firebolt4. In this release the API to add an entry L2_USER_ENTRY table now supports 4bits of PRIORITY on those devices where the priority field is 4 bits.
SDK-52789	728470	56450_A0	In the previous release an assert was raised when running TR53 on KT2. TR 53(DDR Memory Fill/Verify)is not valid for KT2. This issue has now been addressed by changing the TR rule for TR53 to exclude KATANA2. Instead of TR53 customer shall use TR140 for KT2.
SDK-52805	728606	88750_A0	FE1600: added a extra sleep after soft init and before un-isoalte
SDK-52837	729120	56840_A0 56640_A0 56440_A0 56450_A0 56850_A2	Added new field rx_decap_tunnel to bcm_pkt_t structure. This field determines the type of outer tunnel decapsulation, if any, on the received packet.
SDK-52842		56640_B0	Added external field entry move support on TR3 to manage the field entry priorities appropriately.
SDK-52871 SDK-54669	729527	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	An issue was reported with MY_STATION_TCAM not being correctly programmed for the trunk-based TRILL ports . To address this, trunk relevant fields in MY_STATION_TCAM are now correctly programmed for the termination of TRILL packets.
SDK-52892	622534	56846_A0	In the previous release, bcm_port_fault_get() failed on 1G SFP. This has been resolved.
SDK-52896	716978	56840_A0	Support calculating non-unicast trunk hash destination for TD/TD+/TR3/TD2.
SDK-52921	730103	88650_A0 88650_B0 88650_B1	Add entries using bcm_trill_multicast_entry_add with c_vlan=0 is now supported in the following Trill mode: Trill VL (trill_mode=1) Multicast prune mode does not include VSI (trill_mc_prune_mode=0)
SDK-52942	727724	56334_B0 56334_A0	Bcm56334 10G ports has 2 different macs and both of them show counters increasing when receiving packets. In this release, a fix was introduced to address the issue where incorrect values were being retrieved. Fixed the issue to get 10G statistics only from 10G mac while speed set to 10G. And while speed set to 1G, only get statistics from 1G mac counter.
SDK-52965	730480	56634_A0	In the previous release there was no support for bcmCosqStatOutBytes and bcmCosqStatOutPackets stats in bcm_cosq_stat_set and bcm_cosq_stat_get for Triumph.This issue has now been addressed by adding the support for Triumph.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-53025	729729	88030_B0	fix the issue of interlaken port disabled by linkscan thread on bcm88030 device
SDK-53028		56340_A0	Added code to find valid port block in the given list of port blocks. Previously used macro was checking only the first block in the list. The fix iterates through the entire set of valid port blocks to find the corresponding port block.
SDK-53044		56850_A0 56850_A1 56850_A2	When processing ser fifo, if the block is IPIPE block, we will only set mask upon the pipe, and take the lock of accessing sbs_control. And unlock it before returning in this function.
SDK-53046		88650_A0	In Rx thread, when parsing the packet header, the parsing was supported only for little endian. Now big endianness is also supported.
SDK-53059	730593	88650_A0	VPLS: Add support in PWE2PWE. Example in cint_vswitch_cross_connect_p2p_multi_device.c: function "run" should be called with two PWE ports and type1=type2=2.
SDK-53067	730463	All	"rtag" field removed from bcm_l2_addr .
SDK-53104	720590	56840_A0 56640_A0 56843_B0	Added support for the API's bcm_cosq_stat_sync_get, bcm_cosq_stat_sync_get32 on Trident, Triumph Family, Valkyrie. Similar to bcm_cosq_stat_get(), value returned is software accumulated counter synced with the hardware counter.
SDK-53157	732567	All	Updated API documentation that FP action DoNotCheckVlan cannot be set along with action IncomingMplsPortSet as DoNotCheckVlan is set by default whenever IncomingMplsPortSet action is set
SDK-53203	722629	88650_B0 88660_A0 88670_A0	In 6.4.1 we introduce an optimized way to decapsulate overlay headers (L2GRE and VXLAN) for the case of multicast. So far 2-pass solution was introduced where on the second pass IPMC addresses added to the Tunnel termination database in order to terimnate the IP header in the second pass. Using SOC property: DEFAULT_LOGICAL_INTERFACE_IP_TUNNEL_OVERLAY_MC user can set one global LIF ID for all IPMC termination on the second pass. see cint_l2gre.c for more details.
SDK-53248		88650_A0	We exhibit the various ipmc flows via two main functions in cint_ipmc_flows.c: ipmc_flows_rif_ipmc_enabled() and ipmc_flows_rif_ipmc_disabled().
SDK-53253	731741	56334_B0	In the previous release, SDK read back whole mpls label action table to reuse existing entries when invoking the function _bcm_tr_mpls_get_vc_and_swap_table_index. In this release, SDK adds an option not to reuse entries to address performance concern.

Table 117:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
SDK-53264	733415	`88650_A0`	Allow egress snooping for MIPs with out-LIF on system headers. By default, Arad does not provide any out-LIF information when snooping OAM packets at the egress. To allow this behavior set the soc property `custom_feature_egress_snooping_advanced` to 1. When MIP packets are snooped at the egress, the snooped copy will be prepended with an FTMH and a DSP extension. `FTMH.DSP_EXT_PRESENT` will be set to 1 and the DSP extension will include the out-LIF. The snoop command for egress snooping (up-MIP) will always be 2 and for ingress (down-MIP) always 1. Thus, when changing the snooping behavior by calling `bcm_rx_snoop_set()` with 2 or 1 in the `snoop_cmnd` field, the snooping will be updated for all MIPs in the system. Likewise when OAM frames will be snoop by a MIP at the egress, the snooped copy will always have `FTMH.MCID_OR_OUTLIF==2.` By default only multicast LTM packets are snooped to the CPU. The default behavior may be changed with `bcm_oam_action_set().` Calling this function allows setting a new snoop destination or snooping other types of frames. The calling sequence is as following: 1. Configure a `bcm_rx_snoop_config_t` with the desired behavior (i.e. probability, size, `dest_port,` etc.) 2. Call `bcm_rx_snoop_set()` with the `bcm_rx_snoop_config_t` configured above and the `snoop_cmnd` field set to 1 (ingress). 3. Set a new trap with `bcm_rx_trap_create()` and `bcm_rx_trap_set().` For the latter call, the `snoop_cmnd` field in the `bcm_rx_trap_config_t` struct should be set to 1 (ingress snoop command). 4. Call `bcm_oam_action_set()` with the desired configurations. The destination field in the `bcm_oam_endpoint_action` struct should be set to the trap code from step 3 using the macro `BCM_GPORT_TRAP_SET().` The function `bcm_oam_action_set()` will update the egress snooping configurations to match those configured above for the ingress snooping. An example of this can be seen in the function `mip_egress_snooping_advanced()` in `examples/dpp/cint_oam.c.` Notes: 1. In this configuration only MIP snoop is allowed (snooping MEP packets is not supported). 2.Since there are 2 snoop commands used by all MIPs in the system (one each for the ingress and egress), changing the snoop behavior for one MIP will affect all other MIPs in the system. 2. When the packet gets snoop, the forwarded copy uses forwarding strength 3. If soc property `block_trap_strength_pmf_0/1` is set to lower strength then the packet will not get forwarded. 3. To get the described behavior JIRAs SDK-54865, SDK-54726 should be used as well.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-53292		88650_A0 88650_B0	new soc property - scheduler_fabric_links_adaptation_enable when enabled , the scheduler will take current links' states into consideration when generating credits. (mostly useful in multi stage systems)
SDK-53293		88650_A0	When trying to use the System RED Cint to configure DP discard for multiple VOQs. An error occurs after configuring the first 64 VOQs. This error was due to alloc manager miss configuration. The fix is to Change alloc manager system, red max entities (_DPP_AM_TEMPLATE_SYS_RED_DP_PR_MAX_ENTITIES) from 64 to Multiple NOF Queues by 2. We are multiplying since we have 2 pointer per Queue.
SDK-53319	733446	All 56850_A0 56850_A1 56850_A2	Fixed bcm_vxlan_vpn_create, bcm_l2gre_vpn_create API to replace UUC/MC/BC IPMC index using BCM_VXLAN_VPN_REPLACE, BCM_L2GRE_VPN_REPLACE.
SDK-53323	734007	88030_A0	Instead calculate total good packet using the register RUCA..
SDK-53376		56850_A0 56850_A1 56850_A2	An issue was reported where l3 ip6route show was displaying NEGATIVE free entry values. In this release, the way to calculate the free number and the total number of IPv6 entries has been adjusted to address this issue and also to cover the number of IPv6/64 entries. It depends on bcm_switch_object_count_get to get the following objects: bcmSwitchObjectL3RouteV6Routes64bMax bcmSwitchObjectL3RouteV6Routes128bMax defip_64_free defip_128_free
SDK-53380		88750_A0 88650_A0	The previous issue is that deinit can only be exercised after successfully init, it is not acceptable in case a single CPU controls multiple devices, if one device fails in init, we have to reset the CPU which affect other devices. The current fix is that we support the partial deinit which will dealloc the resource that was alloced in previous init failure, then it can do the normal init which means one device init failure don't need to reset the CPU.
SDK-53405	721824	88650_A0	Scheduling elements prints were added to the gport command. Additionally, "gport count" or "gport c" will print a summary of all gport types count. NOTE: bcm_cosq_gport_traverse was extended to include SEs of all types!!!
SDK-53433	731111	All	In TDM bypass mode, in the FDT - the IRE TDM mask mode is configured by default for not reading from RTP link-integrity, but rather using the RTP reachable bitmap. When calling bcm_fabric_tdm_direct_routing_set() the usage of RTP link-integrity is enabled, without the ability to go back. So static link configuration will not be ignored, and the user can configure active links for TDM bypass mode.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-53444	725754	`56840_A0`	While rebooting the system an additional flap was seen on port during the BCM INIT. This has been resolved In phy init code, by disabling it first in case the attached phy is not an external phy and is not in warmboot procedure.
SDK-53451	735769	`56640_A0` `56640_A1` `56640_B0`	On link up, the pause and MTU max values were not retrieved from config, resulting in all the config values becoming obsolete and default values being programmed. Fix is retrieval of the data from the config and override the default values so that the port values reflect the configured parameters.
SDK-53452 SDK-52881 SDK-48849	722247	`56548_A0` `56546_A0` `56545_A0` `56544_A0` `56542_A0` `56541_A0` `56540_A0` `56545_A1` `56540_B0` `56541_B0` `56546_B0` `56544_B0` `56547_A0` `56545_B0` `56542_B0`	For Apollo2 device, the `L3_DEFIP` memory was partitioned into fixed size giving 2K indexes for v4/64V6 and 2K for 128V6. The change is to partition the `L3_DEFIP` table with the user defined values. User can give any number to change the max number of 128V6 entries and V4/64V6 entries. This user defined partition scheme is already supported for TR3/TD2. user needs to set these soc properties . `ipv6_lpm_128b_enable=1` ======> This sets the new scheme active. `num_ipv6_lpm_128b_entries` = XXXX =====> Number of 128V6 routes Without this configuration, the SDK will set the table with default route tables. 128v6=2048 and v4/64V6=2048
SDK-53482		`88650_A0`	Valgrind is a tool that reports cases where code uses uninitialized data. Currently when Valgrind is run and warmboot is done, some warnings are emitted for usage of uninitialized data. The source of these warnings are uninitialized data that are written to a file. In one case, there was a problem with 64 bit systems that would read bad data -- this was fixed. In the other cases, unused and uninitialized data was always written to the warmboot file. This data written to the warmboot file is now initialized without any implication on code logic.
SDK-53487	736250	`56850_A0` `56850_A1` `56830_A1` `56850_A2` `56830_A0` `56830_A2`	In the previous release, the restriction that the queues in strict priority mode must be in consecutive order on Trident2 family was not documented. In this release, this issue has been addressed by documenting the restriction.
SDK-53506	716783	`56850_A0` `56850_A1` `56850_A2`	In the previous release, when the first strict priority member was a unicast queue, the function `_bcm_td2_sched_check_constraints()` returned *ucmap=1 which was not correct. In this release, this issue has been addressed by setting *ucmap=0.
SDK-53507	736772	`56850_A0` `56850_A1` `56850_A2`	This JIRA is a duplicate of SDK-53600. In which the API bcmPortControlMmuDrain is improved to check the empty state of each nodes and queues after the cells are drained.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-53508	736774	56850_A0 56850_A1 56850_A2	In the previous version, in TD2, when the weights of queues were changed dynamically leaving the schedule mode unchanged, the traffic would be disrupted. In this release, the API `bcm_cosq_port_sched_set()` and `bcm_cosq_gport_sched_set()` have been improved to allow weights to be changed dynamically when the schedule mode is not changed.
SDK-53517		56850_A0 56850_A1 56850_A2	Added ability to support three ALPM profiles to provision different Pivot reservations.
SDK-53556	735811	56640_A0 56641_A0 56850_A0 56640_B0 56644_B0 56850_A1 56850_A2	Index for `COS_MAP_SEL` table was being incorrectly set for CPU as ingress port (0). The higig packets ended up in a wrong queue due to wrong index for CMIC ports the index is retrieved from the soc layer and programmed. Corrected the index appropriately in TD2 and TR3 devices.
SDK-53561		56846_A0 56850_A0	TD+ `L2_ENTRY` table is shared between the two pipelines. Only X-pipe has SBUS access to the shared `L2_ENTRY` table. TD2 has independent `L2_ENTRY_x` and `L2_ENTRY_y` tables, but only access type 4 is supported in memory write operation for these two tables. The combination of `L2_ENTRY` table and access type 2 for Y-pipe has been added into skipped memory list.
SDK-53574	737396	All	Fixed buffer overrun in fall-back implementation of `sal_strncpy.` Note that this implementation is not used in any of the primary system environments such as Linux and VxWorks.
SDK-53584		56850_A0	L2X table is read via DMA manner by default. If DMA fails, the table entries will be read via PIO manner again. If there is a parity error in L2X table, both DMA read and PIO read will trigger parity error reporting. The SER logging feature has been implemented to detect and filter the duplicate parity errors for the customer application.
SDK-53600	737427	56850_A2	The API bcmPortControlMmuDrain has been improved to check the empty state of each node and queue after the cells are drained.
SDK-53602		88650_A0 88650_B1 88660_A0	TDM bypass traffic whose destination is the same FAP usually does not go through the fabric. Forcing of TDM bypass traffic to the fabric can be enabled/disabled using: `bcm_fabric_control_set(unit,` bcmFabricForceTdmBypassTrafficToFabric, 1/0); The current state can be retrieved using: `bcm_fabric_control_get(unit,` bcmFabricForceTdmBypassTrafficToFabric, &enabled);
SDK-53611	737404	56634_B0	CPU can send ethernet packet and higig packet. For local switch disable feature, the register `ILOCAL_SW_DISABLE_DEFAULT_PBM_64` should be configured for CPU port when CPU is sending higig packet,but that is missed in SDK. This issue has been fixed.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-53628		88660_A0	Trill: Internal implementation was changed to remove my nickname duplication in SOC_PPC_MYMAC_TRILL_INFO. Fix does not change Trill multi-homing application
SDK-53630		88650_A0 88650_B0 88650_B1 88660_A0	Tunnel APIs replace: 1.Support bcm_tunnel_initiator_create to replace dip, vlan, dscp, ttl, sip and type of IPv4 tunnel with tunnel_id of tunnel or l3a_intf_id or l3a_tunnel_idx of intf. In case of IPv6 tunnel, dip6 sip6, ttl and type can be replaced. In case of ERSPAN tunnel on ARAD+, span_id and l3_intf_id can be replaced. 2.Support bcm_tunnel_terminator_create to replace tunnel_if, if_class and flags (BCM_TUNNEL_TERM_USE_OUTER_DSCP or BCM_TUNNEL_TERM_USE_OUTER_TTL) with tunnel_id.
SDK-53636	737820	All	Added Level 2 warmboot recovery support for the following switch controls: bcmSwitchUseGport bcmSwitchL2PortBlocking bcmSwitchCallbackAbortOnError
SDK-53656	738788	88650_A0	In Ingress Field Processor, validation of ISQ range in bcm_field_action_add() is incorrect, and as a result does not allow usage of the full range. This is fixed.
SDK-53657	737782	56846_A0	When upgrading from sdk-6.2.5 to sdk-6.3.5 scache space for the differential state was not allocated for few modules. Fixed scache reallocation for RX, NIV, VXLAN modules during warm upgrades.
SDK-53661	737925	56850_A0 56850_A1 56850_A2	TD2 TDM Oversubscription Issue, tx failure with mixed 10G/40G configuration was failing. This has been addressed by updates to the oversub group sorting algorithm.
SDK-53672	739010	All	The validation logic which is valid for SQ/MC was preventing the creation of dynamic queues when the indexes were more than 8 (0 -15). Fixed the code to support dynamic queue indexing as well. The SQ/MC index validation is done for non dynamic ports only.
SDK-53673	738994	56850_A2	Corrected the value of macro definition (BCM_IPMC_RANGE_IP6 and BCM_IPMC_RANGE_PIM_BIDIR).
SDK-53674	739094	56450_B0 56450_A0	bcm_cosq_gport_attach API was not able to allocate non-contiguous queues to support more WRR queues/nodes. A new flag BCM_COSQ_GPORT_SCHEDULER_WFQ has been introduced to support this option.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-53684		88660_A0	DEFAULT BEHAVIOR CHANGE. In Field Processor, when using external TCAM, valid entry priorities values are now limited to the range of 0 to 4194303 (2^22). Additionally, when creating an ACL group in the external TCAM, the user may specify the max entries priorities they intend to use in this group by setting the max_entry_priorities parameter in bcm_field_group_config_t. Indicating the max entry priorities will result in improved control performance of the external TCAM driver. Note that if the max_entry_priorities parameter is set, the valid entry priorities values for the configured group are limited to the range of 0 to max_entry_priorities. Also note that max_entry_priorities parameter is only supported for external TCAM.
SDK-53732	732324	88650_A0	In Field Processor, the user can qualify packets according to the trap-code (bcmFieldQualifyRxTrapCode). The bcm_field_qualify_RxTrapCode expects only a bcm_rx_trap_t parameter, indicating which trap. Thus, it does not support User-Defined traps (since no ID can be specified). Besides, for bcmRxTrapL2Cache Trap-code, two possible set of traps can be qualified: 1. By default, the programmable traps are qualified. Due to their HW value (not divisible by 4), only the 2 first programmable traps are qualified. 2. If the SOC property custom_feature_trap_l2_cache_field_reserve_mc_hit is set, then the 8 Reserve-Multicast traps are qualified instead
SDK-53741	738835	88650_A0 88650_B0 88660_A0	In BCM886XX, the L2 traverse HW allows the definition of flexible rules to traverse and modify the MAC Table entries, including a flexible mask (both on entry key and payload). The bcm_l2_match_masked_traverse is implemented, and examples of L2 traverse can be found in $SDK/src/examples/dpp/cint_l2_traverse.c.
SDK-53757	733995	88650_A0 88650_B0 88650_B1	Clear ipv6 tunnel using bcm_tunnel_initiator_clear() is now supported.
SDK-53763		88660_A0	1. Add support of enable/disable learn functionality in bcm_port_learn_set function for Trill port. 2. Add cint with learning disable for virtual RBridge that receive packet with ingress nickname equal to own virtual nickname.
SDK-53770		88650_A0 88660_A0	Advanced VLAN Edit: Added an example functions for QoS mapping configuration in cint_advanced_vlan_translation_mode.c: qos_default_settings(), add_qos_mapping() and set_qos_mapping().
SDK-53776	739518	88030_B0	Support 4Gbits DDR part, allow row sharing and provide a "TmuAllocDump" to show detailed DRAM usage for bcm88030

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-53794		88650_A0 88650_B0 88650_B1	Replace functionality in L3: Added support to replace intf, mac_addr, vlan, port, qos_map_id and encap_id by bcm_l3_egress_create with BCM_L3_REPLACE flags. Added support to replace intf_array by bcm_l3_egress_ecmp_create with BCM_L3_REPLACE flags.
SDK-53800	739936	All 56450_A0	When bcm_mpls_port_add() was called with new label and BCM_MPLS_PORT_REPLACE flag, a new mpls entry was being added without deleting the old mpls enrtry with old label. This has been fixed.
SDK-53802	740202	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	Updated TSC transcription algorithm to resolve incorrect TDM programming for partial TSC configuration
SDK-53810	739299	All	Background: ========= Whenever SDK performs MDIO write/read operation, the linkscan event has been stopped before the operation and restarted after the operation. The reason behind this, prior to CMICm, software has to stop the linkscan before any MDIO write/read operation as the MDIO controller in hardware doesn't handle parallel access between link and MDIO write/read. Problem: ====== When a port is set to disabled while configured at 100M speed, the link down interrupt from the CMICm is missed. The time between the linkscan event restarts and the interrupt generation from CMICm is not synchronized. It could be due to some race condition. Solution: ====== SDK should be able to perform MDIO write/read operation without stalling active linkscan operation now as it is handled in CMICm hardware (well controlled by HW MDIO controller). Hence, removed the bcm_linkscan_pause() and bcm_linkscan_continue() functions from all the miim read/write calls.
SDK-53822		56850_A1 56850_A2 56850_A0	provided new objects to get the information about a) Max 128B V6, 64B V6, V4 entries for a given configuration b)used 128B V6, 64B V6, V4 entries c) Free 128B V6, 64B V6, V4 entries
SDK-53830		88650_A0 88650_B0	When calling bcm_port_match_add with an egress match, with a valid input but a remote port, the return value should be BCM_E_NONE without any configuration. Instead return value is BCM_E_NOT_FOUND.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-53875	737326	56850_A0 56850_A1 56850_A2	There was no BCM API to control the L2 learning per VXLAN logical port before. Now bcm_port_learn_get/set can be re-used to get/set the L2 learning per VXLAN logical port.
SDK-53876	740022	56850_A0	bcmFieldQualifyMhOpcode and bcmFieldQualifySourceVirtualPortValid qualifiers offsets are updated for Ingress Field Processor to match with regfile (56850). Problem : SDK was unable to use bcmFieldQualifyMhOpcode and bcmFieldQualifySourceVirtualPortValid qualifiers in the Key format - FPF3 in Ingress Field Processor. Solution : bcmFieldQualifyMhOpcode and bcmFieldQualifySourceVirtualPortValid qualifiers offsets are updated for Ingress Field Processor to match with regfile (56850). This is done in Initialization routine of Ingress Field Processor for the Key Format - FPF3 .
SDK-53885	740483	56450_A0	Fixed the issue where entry in EGR_MPLS_VC_SWAP_LABEL_TABLE is replaced when a dfferent MPLS port uses the same VC label but having different properties. With this fix a new entry will be created in the above mentioned table instead of replacing the existing entry.
SDK-53890		88650_A0	Fix of building errors occurred when the Makefile includes KBP flags : + FEATURE_LIST := .... KBP + KBP_DEVICE := KBP_ALG and missing WB flags: - CFGFLAGS += -DBCM_WARM_BOOT_SUPPORT - CFGFLAGS += -DBCM_WARM_BOOT_SUPPORT_SW_DUMP - CFGFLAGS += -DBCM_EASY_RELOAD_WB_COMPAT_SUPPORT
SDK-53894		56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0	Support added for software assisted virtual port L2 flush. In addition, there was an issue where the software copy of the L2 entries learnt on external ESM was out of sync with the hardware,. This is now handled by correctly extracting and updating external L2 entries into internal software copy.
SDK-53907	740307	88650_A0	Fixed alloc manager failure in case PWE is created over LAG. The scenario that caused failure: 1. Tunnel application setup using l3_egress_object on a LAG port. 2. Create a PWE on the tunnel interface
SDK-53912	739785	56850_A0 56340_A0	Added ability to support ETAG(Port Extender VLAN Tag) tunneled mirror.
SDK-53919	740350	All	STG Id - 0 is reserved and used for internal purposes only and VLANs should not be added to this group. Updated the users guide with the above information.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-53926	740455	88650_A0	In Metering HW, rate configuration is performed via a mantissa and exponent representation. An internal SW function is used to convert from a rate to the mantissa and exponent representation, and takes as input the maximal allowed exponent and mantissa. Due to a bug, in some cases this function could return an out-of-bounds value (bigger than the maximal mantissa), which causes an assertion failure when the value is written to the HW. This is now fixed.
SDK-53934		56850_A0 56850_A1 56850_A2	In previous releases, if we configured L3Depth larger than 0, it was not possible that single-wide or double-wide entries could be moved to other banks to free its original space for wider entries like double-wide or quad-wide entry. In this release, single-wide or double-wide entries can be moved to other banks to free its original space for wider entry, and the total utilization of L3 benefits from this.
SDK-53935		56850_A0	In earlier releases, bcm_l2_matched_traverse API call did not have a way to retrieve Static only entries. Code has been added to deal with the action for the STATIC only entry. When the action is for the STATIC only entry, we now set STATIC_BIT in both data and mask fields.
SDK-53940		56850_A2	There is one issue for eye scan extrapolation that yields 1e-0.0 BER. The reason is that the sample points used in the extrapolation are less than 2 and the eye scan function rejects its calculation. Also the sample points are not screened, but to include all nodes, to fit the extrapolation equations. This JIRA is to fix these issues by picking or creating the proper sample points for extrapolation. For the PRBS error count, the read back from the PRBS status register is 2X for some counting modes, but the calculation equation is expected to be 1X in eye scan, while 2X in the eye margin. This JIRA irons out the difference by changing the eye margin to expect 1X as well.
SDK-53946		88650_B1 88660_A0	Important note: in Fiber channel APIs, due to an API change, the user must replace bcm_fcoe_zone_entry_t->vsan.vsan by bcm_fcoe_zone_entry_t->vsan_id, e.g. in bcm_fcoe_zone_add API.
SDK-53952	741900	56450_A0	Resource leakage issue in EGR_MPLS_VC_AND_SWAP_LABEL_TABLE caused by bcm_mpls_port_add() API is fixed.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-53955	740686	56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0	In some configurations, ISM memory buckets greater than what is desired was being allocated. This has been addressed by correctly allocating just what is needed, by checking for boundary conditions.
SDK-53956		88650_A0 88650_B0 88660_A0	Egress compensation can be configured for egress ports using the API `bcm_cosq_control_set` (bcmCosqControlPacketLengthAdjust). When the compensation is configured for port with header type `XGS_DiffServ, XGS_HQoSan` error will occur. Fixed.
SDK-53960		88650_A0	when running on little Endian CPU (gto is big Endian) some field BCM APIs may fail, for example following : create pre-selector egress PMF entry with qualifier bcmFieldForwardingTypeIp4Ucast . The fix was in the internal function `"shr_bitop_range_copy"`
SDK-53961		88650_B0	5.75G support is added for ILKN mode
SDK-53963	741711	56850_A2	Fixed VXLAN/L2GRE tunnel initiator's udp port update functionality.
SDK-53968	740158	56850_A0	Fixed to validate VXLAN and L2GRE VPN during `port_delete` and `port_get` API.
SDK-53972		88650_A0	Petra-B-ARAD system: initialize values correctly for system-headers under Petra-B ARAD system
SDK-53992		56640_A0 56850_A0 56640_A1 56640_B0 56850_A1 56850_A2	When a new V6 prefix group is being created and if the start index falls in paired TCAM the following is being done. 1. Check if the previous prefix group has free entries in unpaired TCAM. If yes, set the start of the new prefix group to be that index. 2. if the previous group doesnt have any free entries in unpaired tcam, try to move entries up. if entries can be moved up, then set start as start - 1 of the next prefix group.
SDK-53993	742520	56450_A0	The `bcm_port_match_add()` API was writing the data into wrong entry in `vlan_xlate` table because the search key did not include the field `source_type=1(sglp).` As a result it was not matching the existing entry. Modified `bcm_mpls_port_match_add()` API to include the `SOURCE_TYPE` field as part of key for adding entry in `VLAN_XLATE` table.
SDK-53994	741664	88650_B0 88650_B1	L3: TTL scope entries were not freed when the RIF is deleted.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-53996	741161	56640_A0	Fixed the problem of wrong tunnel index generation in SDK. The tunnel index was wrongly multiplied by 4 before being written into hardware. it made the tunnel index space to use only quarter entries in hardware. Rest of the tunnel indexes values were out of the limit to be written into hardware and resulted in error. Due to this reason, the capwap tunnels were able to scale only to quarter of full hardware space.
SDK-53998	737239	56800_A0 56334_A0	In previous SDK, the statistics snmpIfOutDiscards returned wrong value on some old devices. This snmp counter was mapped to MMU_CTR_MC_DROP_MEMm, but this memory was not existed in some old devices, so the counter should be mapped to some other memories. This issue has now been resolved.
SDK-54001		All	Show KNET protocol override option in CLI help. Added proper support for protocol override in bcm_knet_filter_get API.
SDK-54004		56640_B0	Added the support code for the new SKU BCM56044.
SDK-54009	739826	88650_A0 88650_B0 88650_B1 88660_A0	In L2, during access of static entries (get or delete operations), the parsing of the MAC entry age field was incorrect, causing an internal function failure. The parsing error is fixed, including the removal of an unneeded HW access.
SDK-54014		88660_A0	In BCM88660, the user can select packets which tries to transplant a static MAC Table entry during learning. Specifically, when a statically inserted MACT entry is matched in the learning lookup, but the Source-Port is mismatched, the entry is not modified. If the user wants to match in Ingress Field Processor such packets, The fix includes: 1. Setting Out-LIF valid bit when inserting a static L2 entry with no OutLIF and no valid EEI. 2. Modifying the is-dynamic-entry indication to fix the transplant indication that arrives to the FP. 3. Running cint_field_drop_static_sa_transplant.c (new CINT example) to drop such packets
SDK-54015		88650_A0 88650_B0 88650_B1 88660_A0	In the policer module, when calling the bcm_policer_create and bcm_policer_set functions, the BCM_POLICER_REPLACE can be used to replace the configuration of a meter, or the template that the meter points to. A cint example has been added that shows how to use the BCM_POLICER_REPLACE flag to change the configuration of a meter. For details see the function metering_replace_example in cint_policer_metering_example.c.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54019		88650_B1 88660_A0	Provide a CINT example for followed improvement. In case of L2GRE and VXLAN bud node, MC packet received from overlay recycle port couldn't be decapsulated correctly at previous release. Now the issue was fixed based on new soc property `default_logical_interface_ip_tunnel_overlay_mc`.
SDK-54021		88650_A0 88650_B0	Bug was fixed in event handling of events with high RMEP indices for Arad A0/B0.
SDK-54031	743203	88650_A0 88650_B0 88660_A0	Configuration of the Credit Discount of ISQs is now possible using: `bcm_cosq_control_set(unit, isq_gport,` cosq, bcmCosqControlPacketLengthAdjust, `header_size);`
SDK-54034	743244	56850_A0 56850_A1 56850_A2	Added bcmFieldQualifySrcNivGport,bcmFieldQualifyDstNivGport,bcmFieldQualifyDstGport Qualifiers. In this JIRA, these new qualifiers are initialized only for TD2. Updated bcmFieldQualifySrcGport Qualifier in TD2 to support Niv source GPORT.
SDK-54035		56850_A2	This JIRA is to fix the port status (ps) command about the speed reporting problem for 11G forced speed modes for the TSC driver.
SDK-54037	739743	All	On certain devices which do not support the blocking of `KNOWN_MCAST` type of traffic a fix has been added to no longer return error. This issue was originally reported on Raven
SDK-54038		88650_A0	In Field diagnostics (mode 3) entries validation is performed for all banks. This causes a segmentation fault because it should only be for banks that their owner is PMF, since the entry management for other owners is not performed by PMF. Fixed.
SDK-54042		88650_B1	In BCM L3 file, the macro `DPP_VRF_VALID(_vrf)` definition included a limitation that was not correct for Arad devices. This caused an error when trying to create a L3 interface with VRF>255. The macro definition is changed to support the Arad's limitation.
SDK-54053	743221	56640_A0 56640_B0	On parity error in MMU counters the hardware was not clearing the entries. As a fix, when the parity error happens we now clear parity status and then clear the entry.
SDK-54055		88650_A0	Trill:`bcm_trill_multicast_entry_get` is now supported.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54058 SDK-53881 SDK-53879 SDK-53878 SDK-53880		88650_A0	The Drop-Precedence (DP) is a value that represents QoS internally. The DP is a 2 bit value that represents the colors green (0), yellow (1-2), and red (3). Today, when a packet was received in the device, and its DP was resolved to 2 (or yellow), the device would change it to 1 (also yellow) when sending the packet. This is a result of an attempt to always represent yellow as 1. However in some cases -- for instance when a packet is sent to the device and receives a DP of 2, and this DP should remain 2 -- this will cause problems for the user. This is now fixed by keeping the DP at the same value instead of changing it to 1.
SDK-54063	743248	56643_A0	XMAC_OSTS_TIMESTAMP_ADJUST accounts for delays during the mac stage. This register was always being programmed to zero. Now, This register will be configured with proper value for non GE ports, and for GE ports this will be configured to zero. The value of this register doesnot matter for GMII/MII speeds
SDK-54064	743921	88650_A0 88650_B0 88650_B1	Resolved schan time out when setting pfc refresh timer. The error is caused by reading non-existent register.
SDK-54067		All	Converted MAID value to network byte order before writing to HW table in order to avoid CCM convergence issues arising due to host processor endianness.
SDK-54072	744057	56850_A0	Updated bcmFieldQualifyDstNivGport,bcmFieldQualifyDstGport Qualifiers to support Niv GPORT at EFP in TD2
SDK-54075		88650_A0 88650_B0 88650_B1 88660_A0	VLAN-compression: Delete correctly global VLAN range, in case of no ports refer to it.
SDK-54083	735871	88650_A0 88650_B0 88650_B1	bcm_l2_addr_add() returns error when trunk tgid is used and more than 256. This issue was due to wrong define max value. We fixed the define value.
SDK-54087	743745	56850_A0	In the previous release bcm_mirror_port_dest_add failed with -18 on NIV ports,This issue has been fixed.
SDK-54088		56850_A0	In earlier releases bcm_l3_init() should clear rh_ecmp_flowset but this function did not work. This has been resolved.
SDK-54092		88650_A0 88660_A0	During Driver initialization, all the meter were initialized to use Meter-profile 0. This was unnecessary and removed, since the HW table was already initialized to zero.
SDK-54093	743673	88650_A0	Ingress packet size limit is set to (16KB-128bytes) if the DRAM buffer size is 512 bytes or higher, and set to 8KB if the DRAM buffer size is 256 bytes.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54100		88650_A0	Different threads (Counter Processor, Rx LOS) were failing due to forbidden access to the device during Warm-boot procedure. In counter processor case exceptional access is allowed. For RX LOS initialization postponed till after the Warm Boot finished
SDK-54114		88650_A0	In Field Processor, the Diagnostics command "BCM> diag field res"displays information on databases and banks. The number of entries per bank and database displayed by the diagnostic was incorrect when the entry (bank 0, line 0) was occupied by this Database. Fixed.
SDK-54115	743990	88650_B1	Template management: Resolved issue in a template allocation mechanism that caused on some cases a crash in bcm.user when resource fails to be allocated. This could happen when asking for more profiles than device capable for example: asking more LLVP profile in bcm_port_tpid_class_set.
SDK-54117		88650_A0	port_enable_set API changed: The API is no longer stop EGQ or disable NBI FIFOs. Instead it drops the traffic in the NIF.
SDK-54131	744562	88650_A0 88650_B0 88650_B1 88660_A0	CINT: Adjust vswitch_p2p_init, qos_map_gport to suit the correct number of parameters when called from cint_qos_l2.c, qos_map_l2_run_with_untagged respectively.
SDK-54148		88650_A0 88640_A0	In BCM L3 file, some errors were returned with a generic "TODO err message" text. All error messages in l3.c file are now meaningful.
SDK-54154		88650_A0	In Field Processor, internal and external TCAM tables shared a limitation for the number of uninstalled entries. This limitation is now separated for internal and external TCAMs, in order to allow better control of limitations and memory allocations.
SDK-54162	744768	All	Fixed VXLAN/L2GRE issue with bcm_vxlan_tunnel_initiator_destroy API associated with same-SIP, Multi-DIP scenario.
SDK-54168	738971	56850_A0 56850_A1 56850_A2	In previous release, parity error occurred at second half memory of ING_L3_NEXT_HOP could not be corrected. This has been resolved by correcting the memory depth of ING_L3_NEXT_HOP in SER.
SDK-54171		88650_A0 88660_A0	Move trill deprecated tests from 88640 devices to a deprecated folder.
SDK-54174	744799	88650_A0 88640_A0 88650_B0 88650_B1 88660_A0	VLAN: Remove an incorrect "entry not found" error when calling bcm_vlan_port_create().

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54183	745284	88650_A0 88660_A0	VLAN: Calling bcm_vlan_port_create with the BCM_VLAN_PORT_CREATE_WITH_ID flag, prompts an API check whether the requested ID is available. There was a bug where this check for valid ID would always check if a FEC ID is free, even though the VLAN port might be a multicast ID or a LIF ID. This bug was fixed, and now every VLAN port created WITH_ID would check the appropriate resource is free.
SDK-54185		All	knetctrl filter show did not display the dest_proto field. Show overrided protocol type when showing knet filter infomation in bcm shell. This has been addressed.
SDK-54186	743815	56850_A0 56850_A1 56850_A2	Added SDK Support of ETHERTYPE key in FPF1 Mode 6 in Trident2 Chipset.
SDK-54191		88650_A0 88650_B0 88660_A0	Changed replace logic in bcm_l3_intf_create. After change, when creating l3 intf for the first time, REPLACE flag should not be added. when creating existing l3 intf, REPLACE flag should be added.
SDK-54192	738575	88660_A0	DSCP/EXP marking when bridging allows the user to change the DSCP value of the IP header or the EXP value of the MPLS header of a packet, even when the packet is only bridged. To perform DSCP/EXP marking during bridging, the device is configured to set DSCP and EXP according to the assigned TC, DP, QoS profile and InLIF profile of a packet (map). Due to a SW bug, when configuring a map with TC 4 .. 7, nothing would be configured. As a result packets that have TC 4..7 assigned to them and that DSCP/EXP should be performed on them will get invalid DSCP and EXP values. This fix resolves this issue.
SDK-54194	745534	56850_A0	An SDK crash issue was reported when trying to call bcm_l2_addr_replace() with > 8K MAC address configured on various of vxlan tunnels. This was resolved by correcting the memory allocation. The system now allocates memory for l2 freezing according to the actual size of SOURCE_VPm instead of 8192.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54195		88660_A0	MPLS tunnel works in two modes: Uniform and Pipe. For Pipe mode, struct bcm_mpls_egress_label_t has two flags: BCM_MPLS_EGRESS_LABEL_EXP_SET, BCM_MPLS_EGRESS_LABEL_EXP_COPY, to distinguish between different MPLS pipe modes. In ARAD Pipe mode supports only BCM_MPLS_EGRESS_LABEL_EXP_SET flag. ARAD PLUS supports global configuration of these settings, which is set using switch control bcm_switch_contro_set(unit,bcmSwitchMplsPipeTunnelLabelExpSet, 1); Default of the behavior is EXP_COPY. The flags should be set in consistency with the global configuration. If BCM_MPLS_EGRESS_LABEL_EXP_SET flag is set but bcmSwitchMplsPipeTunnelLabelExpSet switch control is not called, an error will be generated. The same with copy - If BCM_MPLS_EGRESS_LABEL_EXP_COPY flag is set but bcmSwitchMplsPipeTunnelLabelExpSet switch control is called, an error will be generated. See an example of use in: cint_mpls_lsr.c mpls_pipe_mode_exp_set function
SDK-54202	741184	56240_B0	MMU_INTR_MASK bits for CI0, CI1 and CI2 remain reset (set to 1) based on available memory banks. If number of external banks available is 0 then all for all CI0,1,2 the mask will be set. If it is 1 then the mask will be set for CI1 and CI2. if it is 2 then mask will be set only for CI2 and if it is 3 then mask will not be set for any of CI0-CI2.
SDK-54203		56440_A0 56440_B0	Parity checks will now be turned off for non-existent external DDR memory banks.
SDK-54205	738767	56850_A0 56850_A1 56850_A2	It was reported that small packets will be dropped if ENQ_ASF_HS_OVERSUB_EN hasn't been set for the 40G ports which are in oversubscription cut through mode. The issue has been resolved as below: Add 40G/30G ports with oversub to ENQ_ASF_HS_OVERSUB_EN during init. As no matter the ports are enabled CT or not, these ports can always be in ENQ_ASF_HS_OVERSUB_EN.
SDK-54209 SDK-61547	744936	56340_A0 56342_A0	Issue :- Segmentation Fault was observed when more than 254 Flex Counters were created in VFP region in Helix4. Fix :- Maximum number of Flex counters per pool were wrongly assigned during init. Updated the code with correct values.
SDK-54211	682994	88650_A0 88650_B0 88660_A0	A bug in bcm_oam_action_set() causing certain OAM frames to be erroneously pre-pended with an additional set of system headers in certain situations was fixed.

Table 117:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
SDK-54212	690179	88650_B0	The following bug was fixed: OAM endpoints deletion is not releasing internal allocations in case of insertion of a MEP, calling bcm_oam_endpoint_action_set and then deleting the endpoint. After performing this sequence several times a failure will be returned.
SDK-54213	736267	88650_A0 88650_B0 88650_B1 88660_A0	Add received oversized frame counter (ROVR) value to snmpIfInErrors counter (bcm_stat_get API).
SDK-54215		88650_A0 88660_A0	Added documentation for a traffic example and additional documentation per function in cint_qos.c
SDK-54220	745537	88750_B0	When the CL72 mode is enabled, snake test with external loopback failed on fe1600, fixed.
SDK-54230	741970	88650_B0 88660_A0	1588 Termination: Added support to following 1588 termination classification (in addition to already supported 1588oE, 1588oUDPoIPoE): 1. 1588oUDPoIPoIPoE 2. 1588oUDPoIPoMPLSoE 3. 1588oEoMPLSoE Packets will be identified as 1588 packets regardless the forwarding header: Switching (Ethernet forwarding) Routing (IPV4/ MPLS forwarding) or Tunnel (IP/MPLS) termination.
SDK-54233		88650_A0 88650ACP_A0 88650_B0 88650_B1	Change diag pp DB_LIF_lkup_info diagnostic to print lif information instead of the rif information it was displaying previously.
SDK-54235		88650_A0 88660_A0	Egress VLAN Edit: EVE Operations are processed per packet after an ESEM lookup that yields an Out-LIF with a value up to 64K. ESEM entries for Out-LIFs with value above 32K, produced an incorrect Out-LIF value that in turn processed an incorrect EVE action. The fix enables correct EVE behavior for OUt-LIFs above 32K as well.
SDK-54236		88650_B0 88660_A0	In external Tcam, in the application file kbp.c, compilation warnings may appear due to a wrong return value variable type. The variable type is fixed.
SDK-54246	733382	88650_A0	When calling bcm_oam_init(0), counter_engine_source_0 was used for INGRESS_OAM and counter_engine_source_1 was used for EGRESS_OAM, regardless of the soc property configurations. After the fix, any one of the 4 counter_engine_source_Ns may be used for egress/ingress oam, however if OAM is used, at least on counter engine must be set to EGRESS_OAM and at least one must be set to INGRESS_OAM.
SDK-54253	746153	All	Implemented bcm_field_qualify_data_get API for all devices supporting User Defined Function in Field module. This helps to display qualifier data fed into User Defined Function during Field entry creation.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54262		56850_A0 56850_A1 56850_A2	Using the API bcm_cosq_stat_sync_get() to retrieve statistics resulted in incorrect values for counters that are wider than 32bits. The reason being only the initial 32bits were being retrieved. Now the width of the counter is fetched before retrieving the counter value.
SDK-54263	744368	88650_A0 88650ACP_A0 88650_B0 88650_B1	When replacing PWE using bcm_mpls_port_add api with a new push profile, the old push profile was not freed which might cause resources leak. This issue is fixed. Note that replacing push profile is supported only in case PWE is protected.
SDK-54264		88650_B0 88660_A0	Required changes in SDK in order to support KBP-SDK 1.1.1 for external TCAM.
SDK-54266		88650_A0 88660_A0	Fixed a bug in the bcm shell diagnostic function "diag pp cc". The vlan port information displayed in the diagnostic was missing some fields. These fields will now be displayed correctly.
SDK-54269	746371	88660_A0	OAM: Incorrect CCM interval was previously used.
SDK-54271		88650_A0 88660_A0 88670_A0	VPLS: Up to now, calling bcm_mpls_port_create would always allocate both InLif and OutLif. Current enhancement allocates OutLif resource only in case it's required by HW. In other words, in case of PWE unprotected P2P, OutLif is not allocated and can be used for other applications.
SDK-54279	738771	56850_A0 56850_A1 56850_A2	Some PHYs always set the bit XLMAC_RX_LSS_STATUS.LOCAL_FAULT_STATUS no matter the actual speed the port is running at. This leads to always displaying local faults with the CLI command "port xe", which would confuse the customers when the port is running at speeds less than 10G. Now the local faults will be displayed only if the bit XLMAC_RX_LSS_CTRL.LOCAL_FAULT_DISABLE is clear.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-54307		`56850_A0`	This query contained two questions related to the setting for `num_ipv6_lpm_128b_entries` in ALPM mode. The first question was about setting `num_ipv6_lpm_128b_entries` to 3072 with URPF enabled. This question was answered by Zheng Wang, and it looks like we do not support this configuration. The second question was about confirming the table sizes for l3 routes when varying the settings `ipv6_lpm_128b_enable`, `l3_alpm_enable`, and URPF. It looks like the table was mostly right with a small modification in IPv6 64-bit mode `(ipv6_lpm_128b_enable=0).` 2. Disable IPv6-128(config add `ipv6_lpm_128b_enable=0)` 2-1 Combined mode(config add `l3_alpm_enable=2)` IPv4-32(non-URPF)/(URPF) IPv6-64(non-URPF)/(URPF) 128K/64K 85K/21K 2-2 Parallel mode(config add `l3_alpm_enable=1)` IPv4-32(non-URPF)/(URPF) IPv6-64(non-URPF)/(URPF) 64K/16K 21K/5K
SDK-54309		`88650_A0`	KBP compilation fix for not GTO processors
SDK-54314		`88660_A0`	Add diag counter graphical representation for - `EGQ_RQP_DISCARD_SOP_COUNTER`

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-54322		`88650_A0`	1. In Ingress parser, the support of a single IPv6 extension header parsing is added, where only Hop-by-Hop extension is supported. 2. In Ingress Parser, the custom macro allocation is now dynamic. There are four configurable macros (aka custom macros), that are programmed to identify a header. These custom macros are dynamically allocated according to the enabled features (enabled by soc properties). The following soc properties determine the custom macros in the parser: - `bcm886xx_ipv6_ext_hdr_enable` - new soc property that enables IPv6 header extension parsing, requires two custom macros. - `bcm886xx_fcoe_switch_mode` - enables FCoE, requires two custom macros. - `custom_feature_udp_parse_disable` - UDP custom macro is configured by default, however, if needed it can be disabled by this soc property. Note that if disabling UDP parsing, then VxLAN and 1588oUDP are affected. - `trill_mode` - enables Trill, requires one custom macro. -`bcm886xx_vxlan_enable` - enables VxLAN, requires one custom macro. In the specific case of UDPoIPv4oEth, enabling or disabling VxLAN changes the value of parser object end-leaf, which is used in Trap in case there is an error in the Header size or in case of invalid packet format code. 3. In Ingress Field Group, a new qualifier bcmFieldQualifyExtensionHeaderType is introduced, which refers to the Next Header field in first IPv6 extension header after IPv6 header.
SDK-54323		`88650_A0`	Due to inefficient internal implementation, the allocation manager was taking a lot of time during the warmboot recovery. The implementation was changed to reduce the number of function called during restoration and to accelerate the warm reboot.
SDK-54328	743038	All	When configuring an OAM endpoint, `L3_LOCK` was not being released when an endpoint was in multiple maintenance domain levels. This was corrected.
SDK-54329	735713	`88750_A0 88650_A0`	Due to miss-configuration some corrupted cells not dropped as expected. Fixed.
SDK-54343		`88650_A0`	11.25G ILKN speed support is added
SDK-54344		`88650_A0`	Device bring up fail when more than 191 ports are defined. fixed.
SDK-54346	746652	`56850_A0 56850_A1`	`bcm_l3_cleanup` was causing ASSERT error with L3 Egress Mode enabled, `nh_index` -1 could be used for `bcm_xgs3_nh_del` as array index and eventually could cause array bounds write and break the defensive area of allocated memory. Added `nh_index` parameter check to avoid invalid access.
SDK-54347	738808	All	bcmPortControlFabricSourceKnockout was not documented in BCM SDK manual. Added documentation for bcmPortControlFabricSourceKnockout.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54352	743979	56850_A0 56850_A1 56850_A2	In previous SDK, the COS_MAP_SEL table on TD2 sometimes was constructed incorrectly by bcm_cosq_gport_mapping_set API. The root cause of this problem was that the SDK would use ing_port to generate a index of the COS_MAP_SEL table, but this index was overwritten incorrectly and caused the problem. This issue has been resolved.
SDK-54357		56850_A2	The TX squelch function will be persistent through phy enable (on) function, so mac_loopback with port disabled could be operable with the helps of the above functions.
SDK-54363		88660_A0	PON: Trap packets that are send to PON port 12Bytes were added by egress-editor. 12Bytes padding is now removed.
SDK-54369	747308	56850_A2	Fixed VXLAN/L2GRE Tunnel Terminator State modification during multicast_port_create
SDK-54378		88650_A0	To debug more easily warmboot issues, a SW state dump is available via BCM>diag ssdump The SW state dump output to screen can now be disabled.
SDK-54385	747110	88650_B1	In the HW implementation of the Exact Match (EM) tables, a defrag machine can be enabled for all the EM tables. This machine was enabled only for Large-EM, and it is now enabled by default for all the EM tables.
SDK-54395		88650_A0 88660_A0	Support binding 32 LIF cos profiles to InLIF in case of local switching enabled.
SDK-54398	746146	56854_B0 56850_A2	In previous releases, when one interrupt was raised rather than CHIP Function's, only it would beprocessed and the CHIP Function's was lost. In this release they will be processed one by one through comparing all the irqState with irqMask.
SDK-54400	746935	56850_A0	Fixed EGR_PORT_TO_NHI_MAPPING during multicast egress object destroy
SDK-54414	739326	56640_B0	following phy diag command is created to be able to poke into core0,1,3 and MLD register for100G plus port. And the format is phy diag pbm reg core0(core1, core2, mld) aer reg_addr (for read) phy diag pbm reg core0(core1, core2, mld) aer reg_addr write_value (for write)
SDK-54420	746955	All 56850_A0	Only physical gport type supported in function bcm_l2_addr_delete_by_vlan_gport_multi for specific usage, added support for trunk gport type accordingly.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54423		`88650_A0`	Vlan Translation: a new feature is added to support configurable VLAN translation for IP packets according to 5-tuples (DIP, SIP, IP-next protocol, TCP/UDP src port, TCP/UDP dst port). In SW, the sequence to enable the feature is as follows: 1. Set SOC property: `vlan_translation_match_ipv4.` 2. Set VT port profile via `bcm_vlan_control_port_set` API using bcmVlanPortPreferIP4 attribute. 3. Create VSI and add ports to VSI (create InLif). 4. Create Field Group using `bcm_field_group_create()` set QSET with bcmFieldQualifyStageIngressVlanTranslation and all 5-tuples qualifiers. 5. Configure ASET with bcmFieldActionIngressGportSet action, and call `bcm_field_group_action_set().` 6. Add entries to created field group. This feature cannot coexist with EVB support A new CINT is added for example: `cint_field_flexible_qinq_example.c`
SDK-54424		`56850_A0`	Added new shell module - cosq. currently supported sub modules are compensation get/set for ingress/egress
SDK-54426		`88650_A0 88660_A0`	BFD doesn't work properly on management system (one CPU that controls more than one device).
SDK-54429		All	Added new API `bcm_stat_clear_single()` to clear a single port stat
SDK-54435		`88660_A0`	Important note: SOC property `ipmc_vpn_lookup` was misused in code. Default value of soc property was set to 1 but the actual SW implementation is default value 0. Default value of `ipmc_vpn_lookup` changed from 1 to 0 to match SW implementation.
SDK-54436		`88660_A0`	Support to enable global IPMC function when `ipmc_vpn_lookup_enable=0,` and IPV4 compatible MC packets forwarding is according to <RIF,G,SIP> regardless the VRF value.
SDK-54438		`88650_A0`	Added diag for header size difference on ingress and egress. usage: cosq comp ing voq=<id> - show ingress compensation cosq comp egr port=<id> - show egress compensation cosq comp ing [voq=<id>] Compensation=<value> - set ingress compensation cosq comp egr [port=<id>] Compensation=<value> - set egress compensation if only compensation value is give (without port or voq), then all ports/voqs are set with the given compensation value.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54441		88650_B0 88660_A0	OAM ARAD+ RDI can not be set by user. It is updated automatically according to: 1. Scanner LOC discovery on RMEP with same index as the MEP. 2. Received packet information. The mode can be set in bcm_oam_endpoint_create using the following flags: BCM_OAM_ENDPOINT2_RDI_FROM_RX_DISABLE, BCM_OAM_ENDPOINT2_RDI_FROM_LOC_DISABLE
SDK-54442 SDK-54129 SDK-54128		88650_A0 88660_A0	The meter feature has two possible modes of operation - 32K or 64K. In 32K mode, each packet has up to 2 meters with an ID spanning from 0 to 32K-1. In 64K mode, each packet has just 1 meter, with an ID spanning from 0 to 64K-1. The default Meter-ID is 0: - In 64K mode, Meter-ID 0 is set as an invalid pointer. Thus, the meter processor does not perform metering on a packet if its Meter-ID has not been modified. - In 32K mode, Meter-ID 0 was not set as invalid. Since meter 0 is defined to allow the maximal rate, there was no issue with traffic loss. However, it was affecting the color (drop precedence) given to the packet at egress. E.g., if a packet was yellow or red, its color could change to green, ignoring the incoming color, even if a valid Meter-ID was not set to this packet. Meter-ID 0 is now invalid also in 32K mode. Thus, a packet with default Meter-ID will not have its color changed by metering. Additionally, for backward compatibility sake, a SOC property is available to configure the device to set meter pointer 0 as valid: set the SOC property custom_feature_meter_pointer_0_enable to 1.
SDK-54460		56850_A0	In earlier releases, Embedded NH's MAC and Port information was absent in l3 table traverse. This has been resolved.
SDK-54484	745674	56850_A0 56850_A1 56850_A2	BCM_L2_REPLACE_MATCH_UC and BCM_L2_REPLACE_MATCH_MC are provided for specifying which type of MAC entries will be performed the delete operation. Using the BCM_L2_REPLACE_DELETE flag and BCM_L2_REPLACE_MATCH_MC or BCM_L2_REPLACE_MATCH_UC or both to delete all Unicast entries, Multicast entries or both respectively. Using the BCM_L2_REPLACE_DELETE without either BCM_L2_REPLACE_MATCH_MC nor BCM_L2_REPLACE_MATCH_UC is the same as both are set.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54500		88660_A0	In FCoE module, the NPV switch support is added. To configure it: 1. Set the relevant ports to be N_Port by using the bcm_port_control_set API with type bcmPortControlFcoeNetworkPort. 2. Add new routes for source routing by setting in bcm_fcoe_route_add API the flags to (BCM_FCOE_SOURCE_ROUTE | BCM_FCOE_HOST_ROUTE). Refer to cint_fcoe_route.c (fcoe_fcf_npv_example function) for configuration example. When setting the NPV functionality, 2 new FLP programs are required.
SDK-54501 SDK-51080		88650_A0 88660_A0	The template management is an internal module managing the profiles according to their attributes. A diagnostic has been added to display: 1. The profile-ID range per template IDs 2. How many objects are pointing to each profile 3. The raw content of each profile
SDK-54505		88650_B1	OAM packets of all opcodes trapped to the CPU at the egress (up-MEPs) will include the OAM-ID on the FHEI. In 6.3, to attain this behavior the soc property custom_feature_oam_upmep_oam_id_on_fhei should be set to 1.
SDK-54509		88650_A0	In general, SW state must be handled per unit, since multiple device SDK can run on the same CPU. Multiple global SW states have been found not to be defined per unit. Fixed.
SDK-54511		56850_A0	Changed the error type to Parameter error (BCM_E_PARAM) from BCM_E_UNAVAIL for the invalid relative offset input parameter in bcm_field_data_qualifier_ethertype_add() API.
SDK-54515		88660_A0	DEFAULT BEHAVIOR CHANGE (ARAD+ only). Out AC: Out ACs can be created in pairs by calling bcm_vlan_port_create with a BCM_VLAN_PORT_WITH_ID flag and pairs of vlan_port_id. A problem occurs when creating a pair of Out ACs (15 MSBs) with the odd entry created first.When the even entry is created second, the odd entry gets corrupted. The issue detailed above affects the Out AC creation. Pairs of Out AC can be created correctly after the fix in any order. The fix include changing by default all empty EEDB entries to be with bit 34 set to 0.
SDK-54519		56850_A0 56850_A1 56850_A2	In the previous release, hash bits were not being calculated in soc_td2_l2x_hash() function. This has been fixed.
SDK-54529		88650_A0 88650_B0 88660_A0	OAM RDI clear event does not generate a callback
SDK-54533 SDK-57729		56340_A0 56340M_A0	Added separate Ingress Qualifier Init routine for Helix4 device with required offset changes as per Regfile
SDK-54536		88030_A0	It is not necessary to guard against oversubscribing the fabric

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-54544	748071	56840_A0 56440_A0 56850_A0 56450_A0	We received a customer request to make an l2 entry not routable by resetting the flag BCM_L2_L3LOOKUP. In the previous release, some chips in the Trident family did not remove the entry from my_station_tcam table when the flag BCM_L2_L3LOOKUP was cleared. This has now been resolved.
SDK-54545	741393	All 56850_A0 56850_A1 56850_A2	There was a bug in the SDK when configuring the port using the following function in trident2 platform bcm_port_control_set(unit,port,bcmPortControlExtenderType,BCM_PORT_EXTENDER_TYPE_SWITCH) The SDK was setting the PORT.VT_KEY_TYPE_2 field correctly, but was not setting the PORT.VT_PORT_TYPE_SELECT field. However PORT.VT_PORT_TYPE_SELECT_2 field should be set. Fixed the port configuration in function 'bcm_port_control_set'
SDK-54551 SDK-50401	747647	56850_A0	Support has been added for TD2 for bcm_port_subsidiary_ports_get API.
SDK-54557	742238	88650_B1	A priority list is a data structure that keeps a list sorted according to some priority. This data structure is used for TCAM management of the TCAM entries according to priority. Due to a SW bug, invalid memory is read and returned in a local function when trying to get the previous element of the first element, which can cause an invalid memory access. This fix resolves this issue, by returning the head in the aforementioned case.
SDK-54567	748978	88650_A0 88650_B0 88650_B1	The bcm shell diagnostic command "diag pp dblif" support: - for vxlan: key: vni, data: vsi - for l2gre: key: vsid, data: vsi.
SDK-54571	749766	56643_A0 56644_A0 56643_A1 56644_A1	Issue observed was P_START_SPRI was not programmed correctly and this was evident in CLI output of LLS command where the FC "first child" calculation does not match the index of first SP child. Fix provided - in port sched dynamic mode, the P_START_SPRI is correctly configured to the index of first SP child.
SDK-54573	745949	88650_A0 88650_B0 88660_A0	bcm_port_tpid_class_get() should call the SOC_PPD_LLP_PARSE_INFO_clear before using the SOC_PPD_LLP_PARSE_INFO structure.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54585	742690	All	During VPLS Virtual routing using Ingress Field Processor, `REDIRECT_TO_DVP` Action [bcmFieldActionRedirect] in Field module takes Virtual Port information from `ING_DVP_2_TABLE` for devices like Trident, Trident2, Triumph3. Currently we were configuring `ING_DVP_TABLE` only with next hop entries and support to configure `ING_DVP_2_TABLE` was missing. Hence added code to configure next hop entries in `ING_DVP_2_TABLE` during VP port add , through this JIRA. Also added code to delete next hop entries in `ING_DVP_2_TABLE` during VP port delete.
SDK-54589	749529	`56850_A0 56850_A1 56850_A2`	Offset state was not being properly cleaned up when programming flex hash. In this release we have corrected the `UDF_CONDITIONAL_CHECK_TABLE_RAMm` configuration flow when destroy a flexible hash entry.
SDK-54604		`56450_A0 56440_B0 56450_B0`	In previous releases bcmCosqControlBandwidthBurstMax and bcmCosqControlBandwidthBurstMin could not update refresh rate based on burst and shaping rate.The implementation has been modified to calculate refresh rate and update shaper configuration.
SDK-54605	735909	`56640_B0`	In single lane or dual lane mode, if the autoneg is enabled, firmware mode 0 should be used. However all the 4 lanes of that core firmware was to set to 0 instead of relevant lanes only. This has been fixed.
SDK-54606	651774	`56850_A2`	The supports for per-lane PHY controls have been added in TSCMOD.
SDK-54610	750318	All	The counter thread could end up in a continuous loop when sbusdma was busy/not initialized, In this release we have added timeout to break from this loop.
SDK-54615	748837	`56224_B0 56224_A0`	Background: ========= `bcm_vlan_translate_add(),delete, egress_add` and `egress_delete` functions were not implemented to handle wildcard port parameter. Problem: ====== `bcm_vlan_translate_add(),delete, egress_add` and `egress_delete` functions were throwing error when wildcard port parameter was passed. Solution: ====== Added port specific check in `bcm_vlan_translate_add(),delete, egress_add` and `egress_delete` functions which allows user to perform vlan operations by passing wildcard port parameter.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54617	777127	56640_A0 56634_A0 56640_A1 56640_B0 56634_B0	In previous release, the schan response type for devices with ISM, e.g. Triumph3 and Helix4 is not properly checked. The following response types SCHAN_GEN_RESP_L2_MOD_FIFO_FULL, SCHAN_GEN_RESP_MAC_LIMIT_THRESHOLD and SCHAN_GEN_RESP_MAC_LIMIT_DELETE have been added in schan response type checking in the routine soc_mem_generic_insert().
SDK-54619		88650_B0	Added diag "cosq flush" to flush all egress queues per port
SDK-54620		88660_A0	In L2 bridging, the number of MACT entries can be limited globally, per FID or LIF. During packet SA learning, events are sent to the OLP when the MACT entry number limit is exceeded. In BCM88660, a new functionality allows to disable sending these limit-reached messages, reducing the number of created events, by setting the switch_control bcmSwitchL2LearnLimitToCpu to 0. In this case, in the HW, an interrupt is enabled to indicate that the limit was reached.
SDK-54621		88650_B1 88660_A0	For a packet performing a Traffic Management (TM) processing, the regular egress processing removes the system headers (i.e. FTMH and its extensions if exist, PPH and its extensions if exist, User-Header). An improvement allows to remove only the FTMH header and its extensions if exist. If the custom_feature_otmh_keep_pph_<port> SOC property is set, then on this port all the headers starting from the PPH header are preserved.
SDK-54625		88650_A0	increased the table size for warmboot in arad.soc in order to have enough storage space when running with OAM application.
SDK-54635 SDK-37263		56846_A1	In the previous release, SDK only configured the mac driver of current mode when invoking the mac_control_set() function. In this release, we will do mac_control_set() in both XMAC and UniMAC MAC driver except for some special cases. This has been fixed.
SDK-54638	727800	56640_A0 56643_A0 56640_A1 56643_A1 56640_B0 56643_B0	An issue was reported where external FP failed to qualify IPv6/TCP-IP packets with given L4SrcPort and L4DstPort. Corrected the offset of the qualifiers L4SrcPort, L4DstPort for external FP (_FP_EXT_ACL_L2_IPV6 ACL) during qualifiers init.
SDK-54640		88650_A0 88650_B0 88650_B1	For a TRILL Multicast entry, the get/delete APIs did not check both port and MC group match. It could cause deletion of an incorrect entry. This is fixed

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54641	746928	88650_A0 88650_B0 88650_B1	STG: A "STG" diag cli command is added to operate or display STG info of device. The usages of "STG" command are listed as below. BCM.0> stg Usage (STG): Usages: stg create [<id>] - Create a STG; optionally specify ID stg destroy <id> - Destroy a STG stg show [<id>] - List STG(s) stg add <id> <vlan_id> [...] - Add VLAN(s) to a STG stg remove <id> <vlan_id> [...] - Remove VLAN(s) from a STG stg stp - Get span tree state, all ports/STGs stg stp <id> - Get span tree state of ports in STG stg stp <id> <pbmp> <state> - Set span tree state of ports in STG (disable/block/listen/learn/forward) stg default [<id>] - Show or set the default STG
SDK-54642	750484	88230_C0	1) Changed #if/#else/#endif comment at #endif to match #if, which was changed from BCM_FE2000_SUPPORT to BCM_SBX_SUPPORT. 2) Changed several internal functions beginning with string_to to static functions to make the more unique to the specific source file.
SDK-54646		56340_A0	SOC_EGRESS_METERING_LOCK is not unlocked on exceptions which led to crash on event processing. Fixed in the exceptions to unlock the semaphore.
SDK-54661	750105	88230_C0 88230_B0 88230_A0	Fixed Make procedure for 88230 devices
SDK-54669 SDK-52871	787225	56850_A0	Previously, trunk based MY_STATION_TCAM was not programmed for VXLAN and TRILL. Now it is programmed as l3 egress object is created.
SDK-54672	749143	All	Issue :- While doing warm boot(level 2) two times with intra slice double wide group, virtual map information in fp was not recovered properly after the first warm boot and this downgrades the recovery level from level 2 to level 1 during second warm boot. Fix :- While doing level 2 warm boot, after the warm boot succeeds, recreate the virtual map information based on the group information that was recovered.
SDK-54680		88650_A0 88660_A0	MPLS: bcm_mpls_port add() supports BCM_MPLS_PORT_REPLACE flag to replace egress label only if the tunnel port id is protected. A problem occurs when calling bcm_mpls_port_add() with valid egress label but with BCM_MPLS_PORT_REPLACE. It doesn't return fail. bcm_mpls_port_add() returns BCM_E_UNAVAIL if the parameters include a valid egress label but with BCM_MPLS_PORT_REPLACE.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54688		56846_A0 56845_B0 56845_A2 56844_A0 56850_A0 56855_A0 56843_B0 56854_B0 56854_A0 56850_A1	Issue :- A) Mirror resources(Entries configured in im_mtp_index/ em_mtp_index using MirrorIngress/MirrorEgress fp actions) are not cleaned while deleting FP entry in case FP installation FAIL case. B) Only 3 FP mirror actions are allowed even though there are 4 mirror indexes available in im_mtp_index/ ex_mtp_index. Fix :- ------ A) Added Support to clean up the Mirror Resources as well along with FP entry delete in case FP installation failure. B) It is a hardware limitation where only 3 FP mirror actions are allowed.
SDK-54689		88660_A0	Fix for 88660 egress multicast traffic getting stuck in high egress multicast bandwidth.
SDK-54692	747803	88650_A0 88650_B0 88660_A0	OAM: Deleting a MEP with Long MEG ID fails with assertion.
SDK-54711		88650_B0 88660_A0	User-Header is a fabric header located between system-headers (FTMH, PPH) and start of packet (e.g., Ethernet). The user-header size is set via field_class_id_size SOC property. User-Headers-0/1 can have a total sizes of 0, 8b, 16b, 24b or 32b. The value of 24 bits was not enabled. This is fixed.
SDK-54715	722160	88660_A0	In metering, color blind meters are used to do metering without referring to the color of the incoming packet. Currently when a meter is configured to be color blind, it will always drop incoming red packets by error, instead of ignoring the color. This fix corrects this behavior.
SDK-54722		88650_A0 88660_A0	In Field Processor, when creating or destroying TCAM entries, a time consuming debug code section was running. This code section has been removed, resulting in significant decrease of TCAM entries creation and destroying running time.
SDK-54725		All 56850_A0	Support added in 'bcm_l2_addr_delete_by_vlan_gport_multi' API to flush L2 entries based on virtual ports, deletes based on virtual port trunks are also supported.
SDK-54726		88650_A0 88660_A0	The CCM and Loopback programs in the egress PRGE loaded LFEMs that were not used.
SDK-54731		88650_A0 88660_A0 88670_A0	The error message macros in the soc layer were renamed as following: _SOCDNX_SAND_IF_ERR_EXIT --> SOCDNX_SAND_IF_ERR_EXIT _SOCDNX_EXIT_WITH_ERR --> SOCDNX_EXIT_WITH_ERR In addition, a new macro was added: SOCDNX_IF_ERR_EXIT_MSG

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54748	749898	56450_B0 56440_A0 56450_A0 56440_B0	bcm_port_rate_egress_set API allowed user to configure only the recommended minimum burst value irrespective of the passed burst argument. The API implementation has been modified to compute the shaping parameters based on shaping rate and burst value and configure the hardware tables accordingly.
SDK-54755		88650_A0 88660_A0 88650_B0 88650_B1	PON: SDK now supports also L3 subnet source-bind. For more information see src/ examples/dpp/pon/ cint_pon_general_anti_spoofing.c
SDK-54761	752348	All	Fixed potential endless loop during PCIe Deemphasis settings, by limiting the range to search for PCIe Capabilities registers to valid range.
SDK-54763	751831	56450_A0	Support has been added for following features for Katana2: 1. Advanced URPF lookup where 2 lookups, both DIP and SIP, are performed using single L3_DEFIP entry at line rate. 2. Capability to add IPv6 LPM entries with subnet mask greater than 64 bits. By default 1K entries are reserved for Ipv6 LPM entries with subnet mask > 64 bits. The default behaviour can be overridden by setting config variable num_ipv6_lpm_128b_entries=0
SDK-54775	750966	88650_A0 88650ACP_A0 88650_B0 88650_B1 88660_A0	L2CP (Layer2 Control Protocol) traps were not updated correctly when calling multiple times bcm_l2_cache_set.
SDK-54776	751147	88660_A0	The OAM and BFD applications are using TCAM HW to identify some OAM packets on transit tunnels, what causes them to be trapped. A SW bug was allowing using a prefix for this key, and multiple Databases were created (for specific forwarding-types). The fixes are: 1. when OAM is enabled, all the packets performs a look-up into this Database at the forwarding stage (i.e. for any forwarding type). 2. since the HW key length for this TCAM Database, there is no place for prefix and this Database is using exclusively now the TCAM banks 12 and 13. 3. the different Databases were unified to a single Database, since forwarding-type is part of the key.
SDK-54779	748470	56850_A0 56850_A1 56850_A2	In the previous release, the feature of cosq warmboot in TRIDENT2 was not supported. In this release, this issue has been addressed by syncing the left members of _bcm_td2_mmu_info[unit].
SDK-54792		56640_B0	On TR3, if EGR_ING_PORTm register is not configured, L3 traffic received on EHG port seen as source mac and destination mac zero on cpu port. Added configuration for EGR_ING_PORT.
SDK-54802	738723	88750_A0	In polling mode, the hardware IRQ mask is always zero. This is TRUE for all devices. Implementation fixed to achieve this.
SDK-54810	752795	56450_A0	Support added for BCM56450 (Katana2) to match 3 MPLS labels in UDF.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54819 SDK-53920	752509	56450_A0 56440_B0	During port shutdown traffic buffered in the queues for the port was not flushed. Implemented queue flush during port shutdown . Implemented the thresholds reset and replay for the flush activity to be completed during congestion scenarios.
SDK-54840	753240	88660_A0	Broad Sync API: implemented ToD get function (bcm_time_capture_get).
SDK-54845	753234	88650_A0	fixed C++ compilation error: added missing "#include <soc/dpp/SAND/Utils/sand_footer.h>" at the end of arad_debug.h.
SDK-54846	752653	88650_A0 88660_A0	Enabled setting the he Packet-TC to Queue-TC mapping for ISQs using bcm_cosq_port_mapping_set().
SDK-54848		88650_A0 88660_A0	CINT: New example for IPv4 routing over 802.1q where the VSI/RIF is explicitly supplied by the user and not determined by the incoming VLAN value. This allows VSI routing interface that straddles several LANs. The example uses In-LIF (per In-Port x VLAN) and Out-LIF (per VSI x Out-Port) to achieve that. Reference: cint_ip_route_explicit_rif.c. In the CINT example a routing scheme with two different Routing Interfaces(RIF) that are based on <Port, VLAN> added.
SDK-54849		88650_A0	IMPORTANT: Injection of TM packet with user define header is not supported on systems which have OAM yet (SDK-57826). Background: user headers are optional internal system headers located after the FTMH and PPH headers (extensions included). The User header can be used for different purposes: - Cascaded ingress egress ACLs, to transmit data from Ingress FP to egress FP - Various work-arounds The user header size is configured via field_class_id_size_X SOC property. Issue: when injecting TM packets with additional headers after ITMH (e.g. PPH or OAM-TS) and if the user headers are used, the user must include the user headers in the packet after the additional headers and before the payload (e.g. before the Ethernet header). Set this mode via the SOC property custom_feature_injection_with_user_header_enable. In this mode, the user header is not added: injected TM packets must be injected with a User-Header with the same size as the configured user-header size (field_class_id_size_X). If the destination port of the TM packets are Ethernet port, the user also must set the custom_feature_user_header_always_remove SOC property.
SDK-54865		88650_A0 88660_A0	issues in snoop APIs: bcm_rx_snoop_get() now returns the same size and probability as the values entered by bcm_rx_snoop_set().

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54869		`88660_A0`	BFD non-accelerated mep is restored incorrectly after WB.
SDK-54870		`88660_A0`	In Field Processor range APIs, internal commands were added to skip Warm-boot on these APIs during Warm-boot validation.
SDK-54877		`88650_A0 88660_A0`	Diag improvement: IPv4 multicast routing table can be displayed from diagnostic shell Diag pp `IPv4_MC`.
SDK-54880		`88650_A0 88660_A0`	Diag improvement: The allocation manager section now displays general information regarding all pools. In addition, support was added to the detailed information options of the IVE/EVE pools. The "hw" option was renamed to "direct".
SDK-54903	751870	`88660_A0`	OAM: in arad plus, packet below the lowest MEP level was not trapped with trap code `error_level` as it should be but was forwarded.
SDK-54906	749980	`54240_C0 54280_A0 54282_A0 54285_C0 54290_A0 54292_A0 54295_A0`	WarmBoot support for BCM54240/5428x/5429x has been added.
SDK-54907		`88650_A0 88660_A0`	Warmboot: The TPID profile stores up to two expected TPIDs for each port. Each profile can be used by multiple ports and should be discarded when no port is using it. Performing Warmboot has doubled the correct number of ports that are attached to each Port Profile. Thus, preventing proper discard of the TPID Profile when no ports are attached to it and eventually causing an init error that may happen after 256 Warmboots. The WB for Port TPID profile now functions correctly.
SDK-54921		`88650_B1`	In Egress Field Processor, when configuring bcmFieldActionStat action, a validation is performed on the data field value. The validation is incorrect, and in case the value is out of range, it will not be identified. This is fixed.
SDK-54923	752947	`88650_B0 88650_B1`	For stacking systems, the KeepAlive application allows the CPU to retrieve the stacking link topology by sending unicast packets from CPU to CPU. The implementation is performing a specific process in the second stacking device when the FTMH.Stacking-Route-History.MSB is set. However, this process should be done only for Unicast packets. This is fixed
SDK-54927	752923	`56450_B0 56450_A0`	In previous releases `bcm_cosq_gport_delete` API could return `BCM_E_TIMEOUT` during congestion scenarios. This issue has been fixed in API implementation by adjusting bandwidth and flush the packets completely.
SDK-54931		`56854_B0`	If there was an error in the internal functions of the ecmp create routines. the software state was not cleared. Made changes to clean the s/w state in case there is some error in internal routines or h/w writes of ecmp creation.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54937	752666	88650_A0 88660_A0	gport shell command shows incorrect voq id (-1) for ingress shaping queues. Fixed.
SDK-54939		88650_A0	In L2 Control Protocol traps, the attributes of the programmable traps and of the Reserved-Multicast traps are saved in the _rx_virtual_traps variable between the RX-trap API definition and the L2-cache setting. This variable was not restored correctly after warmboot because the struct was not saved to external storage. Fixed. ISSU: if upgrading from an earlier version, this data is not restored.
SDK-54945		56340_A0 56344_A0	Big Endian mode has been added to the SDK when using Helix4 with iProc and latest LDK release To build SDK in Big Endian mode, type "make ENDIAN_MODE=BE".
SDK-54947	752756	56440_A0	SDK support for 1588 Transparent Clock is added as part of this JIRA.
SDK-54971		88650_A0	In Field Processor, the cascaded value width is set via bcmFieldControlCascadedKeyWidth. The bcmFieldQualifyCascadedKeyValue qualifier has a length equal to this value. However, the bcmFieldActionCascadedKeyValueSet action had always a constant length of 20 bits. This length is reduced to {4 + cascaded width}, where 4 bits are needed for HW encoding. This improvement can be disabled by setting custom_feature_increased_cascaded_action to 1.
SDK-54980	753002	88650_A0 88660_A0	6.3.4 introduced a new feature called IGMP and Compatible-MC after existing tunnel (VXLAN, L2GRE, VPLS) in ARAD+. See cint_igmp_example.c for application explanation and valid packet flows. In HW it required to enable Second-stage-parsing in order to make the feature work. Second-stage-parsing should be enabled only for MPLS TT programs. By mistake we enabled Second-stage-parsing to MPLS, IPV6 and Trill while the correct configuration should enable it to MPLS only. The issue cause Packet-format-code to be Ethernet instead of IPV6 (or Trill).
SDK-54982		88650_B0 88650_B1 88660_A0	At egress, a new feature allows to maintain the User Defined Headers (UDH) before the packet exits the device, by defining UDH_ETH property. As a result, UDH is stamped pre-pending the packet headers. Enable this port by configuring the following SOC properties: 1. Update the port header type SOC property definition to UDH_ETH for this port: - tm_port_header_type_out_[port#].BCM88650=UDH_ETH 2. Update the User Header sizes according to the field_class_id_size_X SOC property - see its documentation for the acceptable values.
SDK-54984		88650_A0	Fix an error when setting egress port bandwidth (bcm_cosq_gport_bandwidth_set, using GPORT_LOCAL) to low rate relative to other ports.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-54992	753214	56840_A0	Updated Tx packet padding logic in Linux KNET module to properly handle RCPU encapsulation. The previous code could cause Tx data corruption if the padding required reallocation of socket buffer.
SDK-55003		88650_A0	In Rx thread, more internal fields (from FTMH, PPH and their extension headers) are parsed into bcm_pkt_t. A complete description of the parsed fields will be added to the TM User Manual. The parsing is done for ports of type CPU and STACKING.
SDK-55007	752699	88650_A0 88650_B0 88650_B1	Port: ilkn_interface_status_oob_ignore can be used to force ILKN interface status indication. If ILKN interface status indication is forced up after the ILKN OOB interface is enabled, a low number of error message will be sent from ILKN interface. After the fix, ILKN lane and interface status indication will be forced up before the ILKN OOB interface is enabled.

SDK-55023	734742	88650_A0	A new corrective action added at this interrupt handler which checking if the interrupt is cleared every 10ms. The mechanism stops only if the interrupt clear or period of 500ms passed. moreover we are suggesting to use force_unmask option for this interrupt in order to force unmasking the interrupt at the end of interrupt handler. The following is driver reference for this action: uint32* flags; int inter = /*interrupt number*/;
			rc = soc_interrupt_flags_get(unit, inter, &flags); BCMDNX_IF_ERR_EXIT(rc);
			if (value == 0) { SHR_BITCLR(&flags, SOC_INTERRUPT_DB_FLAGS_FORCE_UNMASK); } else { SHR_BITSET(&flags, SOC_INTERRUPT_DB_FLAGS_FORCE_UNMASK); }
			rc = soc_interrupt_flags_set(unit, inter, flags); BCMDNX_IF_ERR_EXIT(rc);

SDK-55026		88650_A0	XGS MAC extender port support 1G extension capabilities when ARAD/ARAD+ is connected to XGS devices to extend 1G capabilities in chassis. Several Ethernet Inport properties weren't configured right for XGS MAC extender port for example: custom macros for Trill header parsing were not set. Fixed.
SDK-55036		56850_A2	ENQ_ASF_HS_OVERSUB_EN is enabled during init for all the ports in TD2 [SDK-54205] hence the ASF_ENABLE_HS_PORT_EP_CREDIT_CHK also should be set to 0 on init.
SDK-55038	754635	All	Add support for customer-supplied call-backs in the Linux KNET kernel module. The call-backs allow another kernel module to modify packet data and meta data before an Rx packet is handed off from the KNET driver to the Linux network stack, or when a Tx packet is handed off from the Linux network stack to the KNET driver.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55067		88750_A0 88650_A0 88750_B0 88650_B0 88650_B1 88660_A0	Added a mechanism to control logging and console messages formats. Please look at file src/appl/diag/bsldnx.c, function bsldnx_cons_init().
SDK-55071		56850_A0	Implemented new data formats (macros) to match on the incoming packets with or without VNTAG/CNTAG/ETAG/ICNM packets. For example: setting BCM_FIELD_DATA_FORMAT_F_VNTAG flag, an entry is created in UDF_TCAM to validate on the incoming packets tagged with VNTAG. Likewise, setting BCM_FIELD_DATA_FORMAT_F_NO_VNTAG flag, an entry is created in UDF_TCAM to validate on the incoming packets without VNTAG.
SDK-55081		56640_A0 56540_A0 56640_A1 56640_B0 56540_B0	Before the code change bcm_tr2_cosq_gport_get function is returning only BCM_COSQ_GPORT_UCAST_QUEUE_GROUP , now code is added so that it returns flags as per the type BCM_COSQ_GPORT_VLAN_UCAST_QUEUE_GROUP(for vlan gport) BCM_COSQ_GPORT_DESTMOD_UCAST_QUEUE_GROUP(for dmvoq gport) BCM_COSQ_GPORT_MCAST_QUEUE_GROUP (for multicast gport port)
SDK-55083	753905	56340_A0 56340M_A0	ISM total calculation was simplified. Previously total was incremented initially and from then, every time when the number of entries were bumped up. Now the increment will be done only when we allocate the memory from a bank to a table.
SDK-55084	753827	88650_A0 88660_A0	Trap PWE TTL=0/1 is now supported: 1) bcmRxTrapMplsTtl0, bcmRxTrapMplsTtl1 traps are now supported. 2) To set trapping PWE packets with TTL<=1 use bcm_mpls_port_t.vccv_type=bcmMplsPortControlChannelTtl. Example can be found in cint_vswitch_vpls.c
SDK-55095		88660_A0	Trill Warmboot: Upon warmboot, Trill init called to HW access as it shouldn't be.
SDK-55101		88650_B0 88660_A0	Required changes in SDK in order to support KBP-SDK 1.2.1 and higher. The changes include configuration of newly used instructions and their transport layer implementation.
SDK-55102		88650_A0 88650_B0 88660_A0	During initialization, the SOC property configuring the OTMH Destination extension has an uninitialized value, instead of being disabled by default. Fixed.
SDK-55107		88650_A0	Trill warmboot: Upon warmboot, trill sw states were not restored.
SDK-55109		88660_A0	ROP transcations failed when using LE CPU. Fix ROP access endianess.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55132	752736	All 56850_A0 56850_A1 56850_A2	Software state and Ref-counts were not maintained across warmboot. Therefore After warmboot, soc_profile_mem_get api would not be able to retrieve the l3_iif_profile entry as the software states/ref-count are reset and not recovered. Added support to recover the l3_iif_profile state during level-2 warmboot. The bitmap for valid L3_IIF entries are stored in scache. After warmboot, The l3_iif entries are read from scache and ref-counts are set for L3_iif_profiles indexes.
SDK-55143		88650_B0 88660_A0	Required changes in SDK in order to support KBP-SDK 1.2.1 for external TCAM are introduced.
SDK-55161		88650_A0 88650_B0 88660_A0	IMPORTANT - API SIGNATURE CHANGE: For better coherency, the Multicast-ID parameter was changed in the bcm_l2_addr_t structure: the l2mc_index variable was changed to l2mc_group. If used, the user must adapt its calling sequence accordingly.
SDK-55162		88660_A0 88670_A0	IP Routing-Over-Overlay (ROO) refers to a set of protocols/applications where the L2 forwarding to the Host/Next-Hop router is not accomplished by simple 802.1q bridging, but by L2-Overlay protocols (VXLAN, etc). BCM8866X supports ROO Host Unicast over VXLAN. See cint_vxlan_roo.c for cint example and Programmer's Reference Guide for more details.
SDK-55167	755011	56455_A0 56640_A0 56850_A0 56640_A1 56640_B0 56850_A1 56850_A2	Problem: PacketRes enumerations getting remapped internally causing data ,mask mismatch during qualifier installation. Solution: Updated code to qualify packet Resolution in below 2 ways: 1) print bcm_field_qualify_PacketRes(0,0, BCM_FIELD_PKT_RES_L3UCKNOWN, BCM_FIELD_PKT_RES_L3UCKNOWN); 2) print bcm_field_qualify_PacketRes(0,0, BCM_FIELD_PKT_RES_L3UCKNOWN, BCM_FIELD_EXACT_MATCH_MASK); This is also documented as valid set of mask values
SDK-55175	749262	88660_A0	Extracting a BCM88660 that is configured to VSC128 cell format mode, caused performance degradation in the system. Fixed.
SDK-55184		56850_A0	Earlier SDK releases did not allow configuring MTU value for vxlan access ports. This release now supports setting/resetting MTU for vxlan access ports through bcm_vxlan_port_add() API.
SDK-55205	751154	56850_A0	ENABLE_1588MPLSf flag is used to enable/ disable encapsulation and decapsualtion for PTP packets over MPLS. Memory validation check is added to avoid crash while accessing memory for chips that donot have this flag. TD2 does not have this feature.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55222		88660_A0 88670_A0	MPLS: When adding MPLS termination label using api `bcm_mpls_tunnel_switch_add`, action `BCM_MPLS_SWITCH_ACTION_POP`, next protocol after MPLS label will be calculated from next header first nibble and not from the Lif table. Flags `BCM_MPLS_SWITCH_NEXT_HEADER_L2`, `BCM_MPLS_SWITCH_NEXT_HEADER_IPV4`, `BCM_MPLS_SWITCH_NEXT_HEADER_IPV6` are not supported.
SDK-55225		88650_A0 88650_B0 88650_B1 88660_A0	BFD creation of accelerated endpoints with remote destination is fixed and now working without configuring any OAMP instances. Irrelevant validation checks were removed as well. Restrictions on `endpoint_id` and `local_discr` fields: 1. In case endpoint is accelerated to the OAMP, endpoint id should be equal to lowest 16 bits of `local_discr`. 2. In case endpoint is accelerated to the OAMP or endpoint type is bcmBFDTunnelTypeUdp, BFD `local_discr` msbs (bit number 16 and above) should be constant for all endpoints. 3. Non-accelerated endpoint cannot be created `WITH_ID`. Also fixed error in creating oam/bfd endpoint with id 4096.
SDK-55229		88650_B0 88660_A0	When using external TCAM, usage of the diagnostics command "kbp print" may have caused a segmentation fault. This happened due to inappropriate use of unallocated memory and is now fixed.
SDK-55243		All	Improved execution time of `bcm_l3_intf_create()` in XGS devices. In addition, removed deadlock with VLAN APIs such as `bcm_vlan_control_vlan_set()`.
SDK-55263	739431	56540_A0 56440_A0 56450_A0	Fix for PTP operation using little-endian host.
SDK-55274	756256	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	Problem: Src/Dst IP6 qualifier was sent to 32 bit EFP qualifier routine `[_field_efp_qualify32]` for TD2 devices which was internally causing the mask to be reset to 0. [since offset width calculation is assuming width to be 32 but actual width is 128]. Since its a 32 bit routine, the last 32 bit part of mask was getting reset here. Solution: Added appropriate checks to make sure that only 32 bit IP address falls into the check and hence mask will not get reset.
SDK-55280	750005	56440_A0	Support has been added for proper reload of `MAC_BLOCK` table during warmboot for BCM5644x devices.
SDK-55283	756559	All	Removed StrataXGS restriction from `bcm_tx_array` documentation that all packets should have same values for Source module, Source port, PFM and Internal Priority as it does not exist now.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55286	755758	`56850_A0 56850_A1 56850_A2`	When L2X table parity error was detected and processed in Y-pipe context, the `acc_type` list for Y-pipe would be iterated to decode memory id via routine `soc_addr_to_mem_extended().` The `acc_type` of L2X table is 4, not in the list for Y-pipe, and this would cause memory decode fail. So the `acc_type` 4 has been added into the list for Y-pipe to fix this issue.
SDK-55288		`88660_A0`	Trill Multi-homing connectivity. Define up to 3 virtual rbridges in system was not correctly supported and leads to memory leak.
SDK-55293	739558	`88650_B1`	In L2 forwarding, when MAC learning mode was centralized, the aging time accuracy has been improved: the aging time is maximal whether the entry has been inserted by this device or not. A SOC property `(custom_feature_centralized_owned)` allows the user to work in previous mode.
SDK-55296		`88660_A0`	OAM: When replacing entries in the in the O-EM 1/2 tables, instead of deleting the entries and then inserting, it is possible to replace the entries in one fell swoop. Previous configuration might have caused packet loss in the time between the deletion and creation of new entry. Likewise `oem1/2_entry_delete()` did not wait for the task to complete before returning. This bug was fixed as well.
SDK-55298		`88650_A0 88660_A0`	When using lag over a stacking system with `number_of_trunks=[512/256/128/64]` packets might be dropped. When a FAP resolves a LAG destination, it passes the packet to the next stacking FAP with the LAG id and part (8bit) of the lb-key. Since only a part of the lb-key passes, the next FAP may conclude a different destination for the packet. As a result, the packet can be sent back to a FAP that already passed this packet, resulting in dropping the packet. This fix makes the FAP pass the packet to the next stacking FAP with the Destination System Port (DSP) (instead of the lag id), so that next FAP(s) will forward the packet according to the DSP and will not need to recalculate the destination. No change in default behavior, the feature is disabled by default. In order to enable this fix on 6.3.7, the following SOC property configuration is needed: `custom_feature_stamp_uc_destination.BCM88650=1`
SDK-55299		`88660_A0`	OAM diagnostics: Lookups are displayed in parsed format (key and result, if found). The relevant command is diag oam lu and the output is for example: IHB OEMA last lookup: Key=0X2002, result=0X60000080 OEMA key: ingress: 0, OAM LIF: 0x1001 OEMA payload: MP profile: 0x3, MEP bitmap: 0x0, MIP bitmap: 0x80, counter index: 0x0 IHB OEMB last lookup: Key=0X1001e , result=0X0 OEMA key: ingress: 0, MDL: 7, OAM LIF: 0x1001, your disc: 0 Not found.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55305		88660_A0 88670_A0	The masks for Vlan gport id and MPLS gport id has been extended from 24 bits to 26 bits.
SDK-55317 SDK-55378		56340_A0	On Helix_4, Bank 0 in every stage is disabled in ISM. This is taken care of while allocating banks for ISM tables, but while configuring the hash_offset for each bank, the disabled bank was not taken into account. Now the number of disabled banks are calculated and is added to the bank number in each stage.
SDK-55323	755643	56850_A0	For Trident_2, when station tcam entries were being recovered during warmboot, they were not being checked for validity. Even blank entries were being counted as valid entries and so, after recovery, the table showed up as full. So, when a new entry was added after warmboot, it returned no resources. This validity check is now added for Trident2
SDK-55326	715274	88650_A0 88650_B0 88650_B1 88660_A0	PON application: ARAD/ARAD+ Ingress parser used to take two global TPIDs to get the tag formats of packets for each port. With this improvement, ARAD/ARAD+ ingress parser can take an additional global TPID as inner TPID, besides the original two global TPIDs, to get the tag formats for each port. Currently only packets with single outer TPID, which is equal to additional TPID, can be parsed as single s-tag or single c-tag. Please see more information in cint_pon_additional_tpids.c
SDK-55329		88650_A0 88650_B0 88650_B1 88660_A0	APIs receiving a bcm_gport_t input argument as a destination, will now work properly when the gport type is MODPORT, and the given module port is not defined in the local device.
SDK-55332	753717	88650_A0 88650ACP_A0 88650_B0 88650_B1 88660_A0	In Ingress Field Processor, the qualifiers bcmFieldQualifyInnerSrcMac and bcmFieldQualifyInnerDstMac can be taken from second or third header in stack (first or second after Ethernet header). In order to indicate which header to consider, one of the qualifiers bcmFieldQualifyIp4, bcmFieldQualifyIp6, bcmFieldQualifyMpls can be used in QSET. If one of these qualifiers exists, then the inner mac will be taken from the third header, otherwise from the second.
SDK-55335	708385	88650_B0	fixed the prbs issue going out the analog part for 8b/10b encoding speed.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55336		88660_A0 88670_A0	Implemented VRRP and multiple mymac termination for Jericho, and added several updgrades for ARADPLUS: 1. Removed the constraint of disabling VRRP when multiple mymac termination mode was used. In the past, when using multiple mymac termination, l3_vrrp_max_vid soc property had to be set to 0, and calling VRRP APIs would return error. Now setting l3_vrrp_max_vid to 0 would be ignored, and calling VRRP apis in this case will have no effect. 2. VSI 0 is now legal to use for both features. Calling one of the APIs with VSI 0 would configure all VSIs with the selected VRID / mac address in one call, instead of looping over all VSIs. It can be used for features like MPLS-TP multicast packets where 01-00-5E-90-00-00 should terminate Ethernet header.
SDK-55339		88650_A0	Slow start mechanism for FMQs (using bcmCosqGportTypeGlobalFmqGuaranteed control) is not functional. Fixed.
SDK-55344		88650_A0 88650_B0 88650_B1	BFD: fields that are only used by endpoints accelerated to the OAMP are configured only for relevant endpoints. Likewise in endpoint_destroy().
SDK-55345		88660_A0	OAM: RDI indication on outgoing packets from the OAMP might be inconsistent.
SDK-55346		88650_A0 88660_A0 88650_B0	OAM: In Arad, all MEG levels 0-7 may be used. In Arad+, level 0 is unavailable by default, however this may be used if the classifier is used in Arad mode - if the soc property "oam_classifier_advanced_mode" is set to 0.
SDK-55347		88650_B1	OAM: For trapped DM packets (both up and down, NTP or 1588), the packet will be prepended with the 4 MSBs of the time (the 4 LSBs appear in the OAM-TS). In other words, the packet format will be FTMH+OAM-TS+PPH+4 time MSBs+packet. To use the old format where there is only the 34 bits in the TS unset soc property "custom_feature_oam_dm_tod_msb_add_enable=0" (1 by default).
SDK-55350		88660_A0	Adjustedcint_system_vswitch_vpls.c to fit PWE/LSP pipe mode.
SDK-55352	756202	All	In the previous release there was a coding issue with the usage of sizeof operator.The object used to calculate the sizeof operation in a function was passed as a value instead of passing it by reference. Hence sizeof operator was returning a wrong value. This issue has now been addressed in this release.
SDK-55353	757018	88640_A0	Bug in counter processor calculation of counter ids from counter set ids was fixed.
SDK-55359	756745	88650_A0 88650_B0 88650_B1 88660_A0 88670_A0	Changed Init sequence prints.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55360	757697	All	In the previous release, on TD2, the double bit ECC error notifications from the EP following init cycles were seen on rare occasion. In this release, this issue has been addressed by initializing all of packet buffers to the value of zero.
SDK-55361	749578	All 56850_A0 56850_A1 56850_A2	There are three commands have been added. l3 nat_ingress show l3 nat_egress add l3 nat_egress show These commands enhancements to the BCM diag shell to both program and show NAT status.
SDK-55362	757471	88030_A0	The code to set and check individual bit fields of the PPE variable is now automatically generated by the tools.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-55375	755943	`56850_A0`	Problem : Below qualifier was not getting recovered and actions were getting recovered as colour specific. Action Parameters were not getting recovered for CopyToCpu and EcnNew Qualifier: DstClassL3 Actions: CopyToCpu,,EcnNew,DropCancel,PrioIntNew,EgressMask,Drop,EgressPortsAdd,SwitchToCpuCancel Solution: Actions mentioned in the list are expected to work this way since action internally sets colour specific actions. Hence after recovery we read from hardware or cache and display individual actions because we cannot confirm if actual action led to these or they were individually configured. Quoting the part in warmboot section in api document where information regarding the above mentioned actions and its behavior is mentioned. "There are some color-dependent actions that may get aliased during Warm boot recovery. For example, the SDK cannot distinguish whether the application added bcmFieldActionDrop or specifically added bcmFieldActionGpDrop, bcmFieldActionYpDrop and bcmFieldActionRpDrop. This is true for all recovery levels." For CopyToCpu, we are passing param0=1 and param1=0, param0=1 -> means that we are matching the `rule_id` param1=x -> x is the `rule_id` value that we are planning to match. This code will internally check if param0=1 . If so sets a field `MATCHING_RULE` as param1 in `FP_POLICY_TABLE.` Now while recovering we check if `MATCHING_RULE` !=0 and then recover param1 as `rule_id` and param0 as 1. Due to this logic, if param0=1 and param1=0, we set `MATCHING_RULE` as 0. When we recover we dont know if this is due to `rule_id=0` or no `rule_id` configured, because default value for `MATCHING_RULE` = 0. [we dont have any hardware fields to save param0 to check if `rule_id` is to be matched or not] This rules out possibility of configuring param1 as 0 with param0=1 if warmboot recovery required. U can configure param1 as 1-127 with param0=1. Code for recovery of parameter of EcnNew Action and for recovery of DstClassL3 qualifier, has been done through this JIRA. EcnNew has a new Field for Triumph3 and Trident2 to keep the value `[G_NEW_ECNf]` which was missed to be recovered. DestClassL3 was not getting recovered due to double wide mode slice number being passed wrongly. These two problems are handled.
SDK-55387	752326	`88650_A0`	Configuring a discrete WFQ weight for a CL (using bcmCosqControlDiscreteWeightLevel0..3 controls) with the same weight already assigned by another element failed. Fixed.

Table 117:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
SDK-55388	756617	88750_A0 88650_A0 88750_B0	The port enable indication might be wrong after warm boot sequence. As a result RX LOS application will not get reliable state of the port an might try to reset the port. Fixed.
SDK-55392		88650_A0	In internal SOC functions related egress port header type setting, beautify the code by introducing #defines instead of hard numbers.
SDK-55396	757120	56850_A0 56850_A1 56850_A2	For TD2 L3_ENTRY table, the case that parity error located in dedicated L3 banks was missed when UFT shared banks are used. Entry index checking for TD2 L3_ENTRY table when retrieving SRAM info via routine _soc_trident2_mem_sram_info_get() has been added for entry indexes in dedicated L3 banks.
SDK-55415 SDK-59514		88650_A0 88650_B0 88650_B1	"g *" command will display MAC regs only once for channelized ports.
SDK-55426		88650_A0 88660_A0	Setting OCB threshold for ingress queues is done with a voq handle, Use this macro to create the relevant gport handle: BCM_GPORT_UNICAST_QUEUE_GROUP_SET. It used to be the case where setting OCB threshold for ingress queues used a voq-connector handle; this is no longer a valid calling sequence.
SDK-55434		88660_A0	In Field processor, at ingress, the Compare operation performs a comparison between the two halves of key-D in second cycle. The comparison first performs a XOR between the two halves and then AND with a predefined mask. The XOR operation is not enabled and therefore the compare result is incorrect. This was fixed.
SDK-55443		88650_A0 88650_B0 88650_B1 88660_A0	PWE: bcm_mpls_tunnel_initiator_create api can be used to update PWE next tunnel used in encapsulation of multicast PWE packets (unicast packets won't be effected). This functionality is available only when PWE is not protected and MPLS tunnel is used by a FEC entry. Example can be found in cint_vswitch_vpls.c switch_pwe_tunnel function.
SDK-55456		56850_A0 56850_A1 56830_A1 56850_A2 56830_A0 56830_A2	In the previous release, flexible counter thread could occasionally report a huge counter statistic when the two hardware counters belonging to two ports which locate at different pipelines rolled over at the same time. In this release, this issue has been addressed by handling rollover for individual pipelines.
SDK-55460		56850_A2	The access type of ING_NEXT_HOP table is defined as 1 per regfile bcm56850_a0. This access type was missed in TD2 Y-pipe list in SER correction routine. The access type 1 has been added into TD2 Y-pipe list in SER correction routine to resolve this problem.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-55464		`88650_A0 88650_B1 88660_A0`	diag nif shell command speed improvement, PHY rate is measured once for each interface, instead of measuring PHY rate for all interface channels. no modifications required in customer applications.
SDK-55470	758460	`88650_A0`	An updating logic happens when creating VLAN port with `BCM_VLAN_PORT_REPLACE` and `BCM_VLAN_PORT_WITH_ID.` If a new key to be added is different from the existed old key, the updating logic removes the old key and adds the new key. An error occurred when the updating logic removed the old key of egress AC for CEP ports. The updating logic compared the new key with an uninitialized old key to check whether the new key is different with the old key. The issue detailed above affected Out AC replacing of CEP ports. The correct egress AC key can be removed after the fix.
SDK-55471		All	=== FOR THE CUSTOMER USING SDK-6.3.X Customer needs to follow below instructions to create new build target. 1. copy `$SDK/systems/user/gto-2_6 $SDK/systems/user/custom-3_10` 2. modify 2 lines in `$SDK/systems/user/custom-3_10/Makefile` override `kernel_version=3_10` platform=myboard-$ `(kernel_version)` 3. copy `$SDK/make/Makefile.linux-gto-2_6 $SDK/make/Makefile.linux-custom-3_10` and modify `CROSS_COMPILE, TOOLCHAIN_BIN_DIR,` KERNDIR appropriately. 4. copy `$SDK/make/Makefile.linux-kmodule-2_6 $SDK/make/Makefile.linux-kmodule-3_10` Customer doesn't need to modify this file. 5. cd `$SDK/systems/linux/user/custom-3_10` && make === FOR THE CUSTOMER USING SDK-6.4.X Customer needs to follow below instructions to create new build target. 1. copy `$SDK/systems/user/gto-2_6 $SDK/systems/user/custom-3_10` 2. modify 2 lines in `$SDK/systems/user/custom-3_10/Makefile` override `kernel_version=3_10` platform=myboard-$ `(kernel_version)` 3. copy `$SDK/make/Makefile.linux-gto-2_6 $SDK/make/Makefile.linux-custom-3_10` and modify `CROSS_COMPILE, TOOLCHAIN_BIN_DIR,` KERNDIR appropriately. 4. cd `$SDK/systems/linux/user/custom-3_10` && make
SDK-55479	739565	`88030_B0`	Note.
SDK-55487		`88950_a0 88750_A0`	Added logging information during initialization.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55495		88650_A0	BFD: bugs that hindered calling bcm_bfd_endpoint_create() with the flag BCM_BFD_ENDPOINT_UPDATE set for type= bcmBFDTunnelTypeMplsTpCc and bcmBFDTunnelTypeMpls were fixed.
SDK-55500	758887	88650_A0 88650ACP_A0 88650_B0 88650_B1 88660_A0	In the cint cint_policer_metering_example.c, the function header_compensation_example used the wrong function to set header compensation. This is now fixed.
SDK-55501		88650_A0 88660_A0	In Field Processor, when creating a new Field Group, it is verified that the key can be allocated with the existing occupation of the program's instructions. However, the verification does not consider the used key bitmap which may indicate that all LSB/MSB keys are used. In which case, the algorithm should disregard the relevant (LSB/MSB) instructions. This is fixed.
SDK-55502	759144	56450_B0 56450_A0	soc_mem_config_set() (is set to sal_config_set() in our local SDK environment with SAL implementation) may or may not be available with customer code. so assert is not considered good idea. If soc_mem_config_set not available and auto_portgroup and auto_polarity_flip is set true, SDK will suggest settings on screen so that end user can re-update config.bcm accordingly. Also made auto generated config variables unit specific (i.e. portgroup_<num>.unit=<lanes)) happens with auto_polarity_flip and auto_portgroup config variables. This is relevant in multi unit setup.
SDK-55515	752139	56640_A0 56440_A0 56850_A0 56440_A1 56640_A1 56640_B0 56440_B0 56850_A1 56850_A2	bcm_port_learn_set is used to control the learning behavior on a port. The learning behavior can be set/modified using this API. This API was not supporting vlan virtual ports previously. Now, support is added to modify learning behavior for vlan virtual ports.
SDK-55518	757054	56634_A0 56634_B0	START_BY_START error interrupt was not being handled resulting in high CPU utilization. Added handler for this error, to clear the interrupt status register when set.
SDK-55524	759557	88660_A0	bcm_port_loopback_get bug fix for ILKN port in 2 Caui+ ILKN mode (BCM 88660)
SDK-55528		88650_A0 88660_A0	OAM: bcm_oam_endpoint_action_set supports new actions: bcmOAMActionUcFwdAsData, bcmOAMActionMcFwdAsData to configure forwarding the packet instead of trapping/snooping. The destination when calling this api with the actions above should be BCM_GPORT_INVALID. This scenario is useful in case of MIP where we should forward the data as is without any special OAM action.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55531		56340_A0	The variables to calculate the tokens are integers (4 bytes) but on multiplying two integer variables results in a much bigger number which cannot be accommodated in 4 bytes of allocated memory. This was impacting the vstorm control feature. Declared a temp variable of long integer (8 bytes) to store the resultant value to fix the issue.
SDK-55537		88660_A0	When doing metering on packets, it is possible to compensate for Ethernet inter-packet gap (IPG) and/or Ethernet preamble by setting the switch control bcmSwitchMeterAdjustInterframeGap to 20. This will add 20 bytes to the packet size for meter compensation calculation. Currently due to a software bug, this switch control is not set, and no compensation is performed. This is now fixed.
SDK-55540		88650_A0 88650_B0 88650_B1 88660_A0	An access to an HW table (EGQ-VSI-Profile memory) was performed with a uint32 variable, although the table width is 33 bits. It resulted in a memory corruption. This is fixed.
SDK-55542	755020	88650_A0 88650_B0 88650_B1	Ring Port: G.8032 Ring-Port can be associated with multiple VLAN-Ports using bcm_port_class_set(). De-associating a VLAN-Port from a Ring-Port where the physical port is on remote device have sometimes left the de-associated VLAN-Port in a state where it can't be reused and failed when referred by VLAN-Port APIs. The issue was fixed, so that remote VLAN-Ports that are de-associated can always be reused.
SDK-55543	759990	88030_B0	EML_144 supported added to tools: INDEX_TYPE_144 LKUP_EML_144 Note that EML_144 can not be mixed with EML_176.
SDK-55559	760422	56643_A1	The new support for the below port configuration has been implemented. Device =56643 Frequency (MHz)= 450 Option = 4 GbE Port Group (XC[12:0]) = 36 x GbE+1 x GbE High Speed Port Gr 1 (WC[2:0])= 4 x XFI High Speed Port Gr 2 (WC[6:3])= 2 x HG[42] + x F.H [42]" AXP Port Guaranteed Bandwidth = 5G
SDK-55560		56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56340_A0 56344_A0 56342_A0 56342M_A0 56340M_A0	add an workaround for TR3 and HX4 on both cases AT_L2_Limit_019 and AT_L2_Limit_042

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55567		All 56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	Problem: `bcm_field_qualify_data_get` was not working for little endian hosts because of ordering of bytes during copy. Solution: If the host is little endian, updated code to swap the bytes in the expected ordering required for further processing.
SDK-55583		56640_A0 56640_A1 56640_B0	Currently, Policers in cascade mode are mapped based on dot1P priorities of the incoming packet. Two new policer group modes are added : bcmPolicerGroupModeIntPriCascade and bcmPolicerGroupModeIntPriCascadeWithCoupling which map the incoming packets to policers based on internal priority. Policers work in cascade mode where bandwidth flows from higher priority to lower priority.
SDK-55600	719068	56449_B0 56445_B0 56440_B0 56447_B0 56443_B0 56441_B0 56446_B0 56448_B0 56442_B0	Fixed crash observed during Level 2 warmboot on BCM56440.
SDK-55604	760276	56224_B0 56224_A0	Issue :- After warmboot, Recovered Entries were being shown as Disabled. Fix :- The Entries were actually recovered properly both in H/W and S/W But code changes to mark the entries are enabled was missing. Added the code change to mark the recovered entries as enabled.
SDK-55615	758680	88650_A0 88650ACP_A0 88650_B0 88650_B1 88660_A0	Counter processor example cint, `cint_voq_count.c,` was updated. The bcmCosqGporYellowAcceptedPkts counter type replaced by bcmCosqGportNotGreenAcceptedPkts. This change reflects change in the counter processor counters in `FULL_COLOR` counting mode from version 6.3.2

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55620	758957	56640_A0 56540_A0	Two issues fixed as a part of this JIRA: Issue 1: Linkscan SW mode becomes NONE after executing the ibod WAR. Fix: IBOD sync recovery function _bcm_tr3_ibod_sync_recovery_port() is invoked from different threads and as the function is not properly protected, there is difference in the linkscan states. Provided the synchronization using IBOD_LOCK. Issue 2: When bcm_port_enable_set is getting called even before the LS thread is updated its bitmap ,so when bcm_port_enable_set calls _bcm_tr3_ibod_sync_recovery_port it takes the snap shot of port mode ,which comes to "BCM_LINKSCAN_MODE_NONE" so later on at the end of the function when it update the port mode it removes it from LS ,that where we see some times port is not part of linkscan. Fix: During the ibod WAR execution, the links of the port are set to link UP forcefully by invoking the API _bcm_esw_link_force() API with flags _BCM_LINK_STATUS_NO_CALLBACK. The flags are introduced newly and if the flag is set, the link state change notification is ignored to the registered linkscan users in function _bcm_esw_linkscan_update_port.
SDK-55621		88650_B1	When replacing existing MTU value using bcm_l3_intf_create api, the MTU value might in some cases change to 0 instead of the requested value. This happens in case MTU value is unique for certain L3 Intf
SDK-55630		88660_A0	OAM: when calling bcm_oam_loss_add() with the flag BCM_OAM_LOSS_SINGLE_ADDED set, loss management will be based on LMM PDUs, otherwise on CCM PDUs.
SDK-55631	758623	88650_B1	It is now possible to assign ports with a vlan translation port property, and create IP tunnel terminators that use {SIP,DIP,Next_protocol,Port_property} as key for tunnel termination. To activate this mode, use soc property: bcm886xx_ip4_tunnel_termination_mode= 4 or 5 For an example, see cint_ip_tunnel_term.c, call ipv4_tunnel_term_next_protocol_example with use_port_property=1.
SDK-55632		88650_B1	In FLP program selection initialization, some program IDs may have been overridden due to static program ID allocation that followed dynamic program ID allocation. For example, there were conflicts between FCoE and MAC-in-MAC FLP programs. All dynamic allocation of program IDs is now after static allocation, so that no program ID override can be caused. Note that if ISSU is performed, the fix will not apply.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55639		`56850_A2`	In earlier releases, nexthop and ecmp reference count were not decreased when replacing vxlan port. This has been resolved.
SDK-55654	754909	`56850_A0 56850_A1 56850_A2`	Fixed DMA abort sequence in KNET Linux kernel module.
SDK-55661	761066	`56548_A0 56547_A0`	Support for F.HG[42] [SDK-46947] has been ported to the `6_3_branch`.
SDK-55681		`56850_A2`	In the previous release, assertion happened when `bcm_cosq_port_mapping_set` was called in ETS mode.In this release, this issue has been addressed by configuring a correct field of `COS_MAPm` and modifying the queue mode of HG ports to the value of zero.
SDK-55683		`53394_A0`	Added SPI slave mode support of BCM56150 family. Only pure register access path is available in this mode without interrupt and DMA and the access speed is pretty slow in comparison to PCIe.
SDK-55691		`88650_B1`	In L3, when calling the API function `bcm_l3_host_add(),` a lock may have been taken but not released in some cases. The lock is always released now before exiting the function.
SDK-55710		`88650_A0 88650_B0 88660_A0`	OAM: Deleting a MEP with RX configurations only (gport field in `endpoint_create` api is `BCM_GPORT_INVALID)` was failing.
SDK-55712 SDK-55535		`88650_A0 88660_A0`	Add the option to Use Dram saved config Parameters, and in case there are no Parameters to Perform Shmoo on init. Set this option as Default. # 2 = Use Dram saved config Parameters, if no Parameters Perform Shmoo on init. Default option. # 1 = Perform Shmoo on init. # 0 = Use Dram saved config Parameters, if no Parameters do nothing. `ddr3_auto_tune.BCM88650=2` Also, as default Load DRAM tuning properties from local File `(/home/negev/bcm88650_dram_tune.soc).` RcLoad will not fail if file not found.
SDK-55713		`88650_B0 88650_B1 88660_A0`	Broad Sync API: implemented all missing `bcm_time_*` APIs .
SDK-55715		`88650_A0`	PWE: verification case of updating TPIDs per PWE using bcm port tpid APIs `bcm_port_tpid_add/delete` does not work correctly (API always update TPIDs regardless of gport type)
SDK-55719		`88650_A0 88650_B0 88660_A0`	OAM: api `bcm_oam_endpoint_get` returns incorrect flags in field flags2.
SDK-55720		`88650_A0 88660_A0`	In Ingress Field Processor, when using TM programs per port profile (soc property `post_headers_size` is set), the program selection shuffle algorithm resets lines of Ethernet programs due to incorrect range calculation. This is fixed.

Table 117:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
SDK-55722	761214	`56450_A0 56450_B0`	External DDR has 1024 columns but SDK was assuming it as fixed 2048. Due to this , calculated max bist address was crossing boundary and SDK was thowring assertion(crash) message. Issue is fixed by below below two steps 1) Added safety check for max BIST Addr. With this, even if user passes wrong parameter for running DDR TR 140 test case, assertion (i.e. crash) will be avoided. 2) Used config properties `(ext_ram_columns` and `ext_ram_banks)` to set `NUM_COLUMNS` and BANKS of external DDR memory. i.s.o. fixed 2048 and 8. Default will be 1024 and 8. Settings will be displayed on scrern. With this, if required , user can changes values based on connected DDR capability.
SDK-55727		`88650_A0 88650_B0 88660_A0`	OAM: Mac-In-Mac OAM packet identification causes non-oam packets to be trapped to OAM engine.
SDK-55730		`56850_A0 56850_A1 56850_A2`	In the previous release, `bcm_td_cosq_gport_detach` intermittently returned `BCM_E_RESOURCE` incorrectly when the schedule nodes were not used up. In this release, this issue has been addressed by releasing the schedulers which are used by legacy setup once ETS mode is enabled.
SDK-55736		`88650_B1`	In FCoE application, a new improvement allows the support for VSAN assignment from VFT or VSI (according to a device configuration) and supports a default VFT value per incoming port. The calling sequence is: 1. Set the default VSAN assignment between VFT (by default) or VSI via `bcm_port_control_set(unit,` port = -1, type=bcmPortControlFcoeFabricSel, value = bcmPortFcoeVsanSelectVft / bcmPortFcoeVsanSelectOuterVlan) 2. If the mode is VFT, set the default VFT per port via `bcm_port_control_set(unit,` port, type = bcmPortControlFcoeFabricId, value); Note: FCoE application cannot co-exist with the usage of the Field Processor bcmFieldQualifyInterfaceClassProcessingPort qualifier at external stage (bcmFieldQualifyStageExternal) due to the usage of the same HW resource (the port key profile in forwarding stage).
SDK-55740 SDK-56736	757357	All	`sand_erorr_code` mechanizm shouldn't be used without initialization. If init sequence failed before initializing the error mechanizm & deinit try to use it Segmentation error will occur. In order to solve this problem we are not using `sand_error` mechanizm at deinit sequence.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55766		56640_A0	Currently, Policers in cascade mode are mapped based on dot1P priorities of the incoming packet. Two new policer group modes are added : bcmPolicerGroupModeIntPriCascade and bcmPolicerGroupModeIntPriCascadeWithCoupling which map the incoming packets to policers based on internal priority. Policers work in cascade mode where bandwidth flows from higher priority to lower priority.
SDK-55770	762574	56850_A0	In previous releases, if multi-thread user accessed L3 memory with different view (i.e., L3_ENTRY_IPV4_UNICAST, L3_ENTRY_IPV4_MULTICAST), the physical memory could be corrupted because different view of same physical memory was using different LOCK, and the entry movement between banks could happen with invalid LOCK protection. In this release, different views of same physical memory are pointed to same LOCK, so the protection is effective.
SDK-55793	757103	88650_A0 88650_B0 88660_A0	VPLS: Enabled modification of working Incoming-PWE configuration under traffic by first creating new instance (the traffic will be moved to the new instance), then deleting the old entries using bcm_mpls_port_delete(). Example can be found in cint_vswitch_vpls.c when make_before_break field is set to 1.
SDK-55803 SDK-55946		88660_A0	When using bcm_port_control_set with the control bcmPortControlEgressModifyDscp, an inlif profile is expected in the port argument. An issue was found when one of the ports 0-16 is disabled. In this case when using an inlif profile with the same number as a disabled port, the API will produce an error, even though the argument is valid. This is now fixed.
SDK-55818	761770	56334_B0 56334_A0	In the previous release, SDK delete old next hop entry before new entry was installed when invoking the bcm_mpls_port_add API with BCM_MPLS_PORT_REPLACE flag asserted. In this release, this has been changed to delete old entry after new entry is installed.
SDK-55822		88650_A0	LUT ROP transcations was failing when using LE CPU. Fix LUT ROP access endianess Also improve KBP code by: Add NULL checks at XPT layer. Add ARAD_KBP_ROP_DEBUG_PRINTS define around prints to Improve access time.
SDK-55823		88650_A0 88660_A0	Function that related to Petra-B in Trill moved to trill.c/h files. Remove initializing of sw-states(mc_trill_route_info_db,mc_trill_root_src_db) from ARAD.

Table 117:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
SDK-55825		88650_A0 88650_B0	IMPORTANT: In Rx parsing the src_gport and dst_gport interpretation and values were switched. Before, due to a bug, the dst_gport had the same interpretation as src_gport. From now on, the src_gport is the Source-Port where the packet enters the device and dst_gport is where the packet exits the device.
SDK-55830	763499	88650_B0 88650_B1 88660_A0	Trill Ingress learning: For TRILL egress MC RBridge, it learns according to the native SA and VSI. The original instruction which is used to learn native SA is incorrect for ingress learning. It caused ARAD to learn a random SA. The correct SA can be learned after fixing the instruction of lookup native SA.
SDK-55831	762481	56340_A0 56344_A0 56342_A0 56342M_A0 56340M_A0	Helix4/Triumph3 supports SW based aging and when age interval was modified, the new value did not take effect immediately. Code changes have been added to notify the SW Aging thread when age interval is updated, to take immediate effect.
SDK-55840	761378	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In earlier releases entries[] can be potentially used without initialization in _bcm_td_cosq_wred_set. This has been resolved.
SDK-55850		56846_A0	Support has been added for HG[11] and force cl72 on TD+.
SDK-55857		88650_A0 88650_B0 88650_B1	IMPORTANT: the interpretation (and value) of pkt->pkt_len has been changed. In Packet parsing, 2 fields in bcm_pkt_t are referring to the packet length: 1. The tot_len (total length) field is unchanged, and corresponds to the packet length as received 2. The pkt_len field is changed to correspond to the packet length without the internal headers (i.e. system header size as FTMH, PPH, etc.). The previous value of pkt_len was equal to tot_len.

Table 117:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
SDK-55859	758730	`56640_A0`	Two issues are resolved as a part of this JIRA. Here is the description: Issue 1. When a port gets the link up notification and also a remote fault is detected on the port, the port remains down and the `linkup_bitmap` and `fault_bitmap` for the port are set. After this state if the port is removed from the SW linkscan mode (the `fault_bitmap` is cleared) and if a link up notification occurred after adding the port back to SW linkscan. The function returns without setting the link status (as `linkup_bitmap` is already set to up) and it causes the SW link status for the port in down status. Solution- While clearing the `fault_bit` map, also clear the `link_bitmap.` So that when the linkscan for the port is enabled, it updates the state in the next iteration as per the new link status. Issue 2: When `bcm_port_enable_set` is getting called even before the LS thread is updated its bitmap ,so when `bcm_port_enable_set` calls `_bcm_tr3_ibod_sync_recovery_port` it takes the snap shot of port mode ,which comes to `"BCM_LINKSCAN_MODE_NONE"` so later on at the end of the function when it update the port mode it removes it from LS ,that where we see some times port is not part of linkscan . Solution- For this particular scenario, while restore the linkscan mode after the ibod WAR, retrieve the current linkscan mode and comparing it with the mode it got set before ibod WAR `(BCM_LINKSCAN_MODE_NONE),` if it is not same, the linkscan mode is not restored.
SDK-55882		`88650_A0`	In Warmboot module, some fixes are inserted to prevent some uncatched `wb_engine` setget timing issue (in ipmc module on 6.3 branch). Besides, the error mechanism in `wb_engine` is changed to raise assertions when uninitialized SW database is accessed.

SDK-55885		`88650_B0` `88650_B1`
		`88660_A0`
			In case of User Defined Header, Egress Programmable Editor default program incorrectly removes some data bytes from the packet. the fix updates the additional `bytes_to_remove` to 0. This way no additional bytes are removed beside the system and network headers.

SDK-55889	762107	`88650_B1` `88660_A0`	In Field Processor, when creating Direct Extraction field group, only one (1) qualifier is allowed to be used as filter qualifier per entry. When calling `bcm_field_qualify_data()` for an entry, and then calling `bcm_field_qualify_xxx()` the operation succeeds when an error should be produced. This is fixed.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55902		56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56340_A0 56640_A1 56643_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0 56344_A0 56342_A0 56342M_A0 56340M_A0	Problem: When REGEX feature is enabled, compilation of SDK for linux kernel mode fails. This is because of inclusion of ctype.h file, which is not available directly. Solution: Removed the inclusion of ctype.h. Also re-define the logic in the function isprint() as a new function local to the file, since isprint() is dependent on ctype.h. Affected platforms: All platforms where REGEX is supported.
SDK-55903		56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56340_A0 56640_A1 56643_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0 56344_A0 56342_A0 56342M_A0 56340M_A0	Problem: When REGEX feature is enabled, the compilation of SDK in Linux Kernel mode fails because of variable declarations mixed up with code. Solution: Moved the variable declarations to the beginning of the function and removed some dead code to get the compilation working. Affected Platforms: All platforms where REGEX is supported.
SDK-55913	763695	88650_B1 88660_A0	OAM may be initialized without setting any of the counter_engine_source_{0,1,2,3} soc properties to EGRESS/INGRESS_OAM. Notice that in this case LM functionality is not supported.
SDK-55915	764134	56850_A2	In earlier releases,inALPM mode, even if we disabled URPF, the bits URPF_LOOKUP_CAMx in register L3_DEFIP_KEY_SEL was still 1 Switching back and forth between urpf and non-urpf could result in URPF_CAM_LOOKUPx bits always set to 1. This has been addressed by making sure register settings are set correctly every time urpf switch control changes, and not just the first time.
SDK-55919	764630	56850_A0 56850_A1 56850_A2	Previously, bcm_vxlan_port_add with BCM_VXLAN_PORT_REPLACE overwrote CML_FLAGS set by bcm_port_learn_set. It is fixed now.
SDK-55920	742940	88030_A0	Fix EML304 and EML424 lookup for bcm88030
SDK-55921	764681	56850_A0	In earlier releases, nexthop and ecmp reference count were not decreased when replacing vxlan port. This has been resolved.
SDK-55935	763171	56850_A0	In earlier releases, the disabled pbmp of flexible ports was not recoverd during the warmboot. This has been resolved.
SDK-55942	764885	56850_A0	Implemented following IFP missing actions on TD2. bcmFieldActionPortPrioIntCosQNew bcmFieldActionRpPortPrioIntCosQNew bcmFieldActionYpPortPrioIntCosQNew bcmFieldActionGpPortPrioIntCosQNew
SDK-55945		88650_A0 88650_B0 88660_A0	Allocation manager malfunction was fixed in OAM and L3 applications. The bug was in allocating new profile resources (oam endpoint new actions, ttl scope) instead of existing profile.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-55956	764773	88660_A0	In trap module, the bcm_l2_cache_set API is used to configure Reserve-Multicast and Programmable traps. This API returns an index, which can be used to delete the trap with bcm_l2_cache_delete. Due to a SW bug, bcm_l2_cache_delete was allocating another trap instead of deleting the allocated one. This is fixed. Besides, bcm_l2_cache_get was returning incorrectly the EtherType (and its mask) parameters. This is fixed.
SDK-55964	742713	88650_B0 88650_B1 88660_A0	VLAN-Port Protection: Replace functionality of 1:1 protected VLAN Port to update failover_id is now available
SDK-55967	755351	88650_B0 88650_B1 88660_A0	OAM/BFD: When calling bcm_bfd_init() after bcm_oam_init(), not all BFD functionalities were properly initialized. Analogously when calling bcm_oam_init() after bcm_bfd_init().
SDK-55968	756702	88660_A0	OAM: configuring correct counter pointer for accelerated loss management, as well as correctly stamping counters on CCM based LM.
SDK-55970		56440_A0	The parity protection on TCAM tables is implemented via SER engine and a SRAM table that is utilized to store parity bits of TCAM entries. Only enabling SER engine for the new-added L3_DEFIP table but not clearing its corresponding SRAM portion will leave the parity bits of L3_DEFIP table in an uninitialized state with random values. if the table is dumped, SER engine will check entry parity bits of table entries, this will trigger many parity errors reported. Besides adding L3_DEFIP table into SER engine protection list, memory clear operation for L3_DEFIP has also been added to initialize the parity bits of L3_DEFIP table into correct values.
SDK-55972	764939	56850_A0 56850_A1 56850_A2	Code for Warmboot support of MPLS_EXP_MAP has been added.
SDK-55974		88650_A0 88650_B0 88650_B1	When using external TCAM, the access ROP mechanism was substantially improved. The following new compilation flags are available: ARAD_KBP_ROP_OPTIMIZATION - enable ROP performance optimization. ARAD_KBP_DISABLE_IHB_LOOKUP_REPLY_FOR_ROP_TRANSMIT - enable ROP optimization without reading the IHB reply registers. ARAD_KBP_ROP_TIME_MEASUREMENTS, ARAD_PP_KBP_TIME_MEASUREMENTS - enable time measurements.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-55997		56640_A0 56544_A0 56542_A0 56541_A0 56540_A0 56524_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0 56524_B0 56540_B0 56541_B0 56544_B0 56542_B0	Enhancement:- There are 16 FP physical tcam slices with 512 entries per slice. There are 8 physical FP meter pools with 1024 entries per meter pool. Currently only 8 physical tcam slices are allowed to access the 8 FP meter pools. Requirement was to ensure the 16 physical tcam slices are allowed to access the 8 FP meter pools Support :- The 8 FP Physical meter pools are split into 16 logical meter pools so that the 16 FP tcam slices can attach to the 16 Logical FP meter pools.
SDK-55998		56240_B0	Support has been added for the new Saber SKUs BCM56245 and BCM56246 with support for 256k buffer entries/192MB buffering.
SDK-56009	765570	88650_A0 88650_B0 88650_B1 88660_A0	In Rx Trap module, an error is fixed when calling `bcm_rx_trap_type_create(unit,` 0, type, `&trap_id)` with 'type' as one of the following : - bcmRxTrapIpv4SipEqualDip - bcmRxTrapIpv4DipZero - bcmRxTrapIpv4SipIsMc
SDK-56013	765696	56850_A2	Fixed `tunnel_initiator_delete` followed by `tunnel_initiator_create.In` previous releases, this case could results in an abort of the SDK.
SDK-56015		88650_A0 88650_B0 88660_A0	OAM: MIPs default behavior was changed to the following: MIPs are transparent to all OAM packet types except for LTM unicast, LTM multicast and LBM unicast. If a MIP receives any other OAM packet with destination address == MIPs MAC address (configured in the `dest_mac_address` field in `bcm_oam_endpoint_create()),` the packet will be trapped to the CPU with trap code oam-error-level. If the destination address != MIPs MAC address then the packet will be forwarded (it was trapped to the CPU until now).
SDK-56017	765489	56840_A0	Enhanced warmboot shutdown to detach and close KNET device when present. This allows the application to remove the KNET kernel module without exiting.
SDK-56022		56850_A0 56850_A1 56850_A2	In the previous release, `bcm_vxlan_port_delete` returned `BCM_E_NOT_FOUND` for default VPN associated NW port. The problem was caused by the flex-counter detachment on VFI table. Originally the detachment was implemented in the deletion of VxLAN logical port and thus the operation on VFI table was executed repeatedly when deleting many logical ports in the same VFI. In this release this issue has been fixed by moving the operation to the VPN destroy.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-56024		56850_A0	There's a bit in the VLAN_XLATE table called VLAN_ACTION_VALID,It must be enabled to process XLATE__DISABLE_VLAN_CHECKS for VXLAN virtual ports, but disable it for VXLAN access ports to drop packets at ingress. They have conflict. To solve the problem, a new flag has been added that allows the customer to control the bit, The new flag is BCM_VXLAN_PORT_ENABLE_VLAN_CHECKS.
SDK-56033	765288	56850_A0	Problem: Multiple Mirror Ingress actions were not removed during bcm_field_action_remove_all because during the action remove routine, we removed first MirrorIngress action and then returned without further processing the remaining actions Solution: Updated code to loop through all the actions [in case of similar group of actions] to remove each one of them in action remove routine.
SDK-56038	766065	56850_A0 56850_A1 56850_A2 56455_A0	During warmboot, the reference count for DSCP_TABLE has been updated to reflect the coldboot state.
SDK-56040 SDK-56095	766058	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In earlier releases bcm_esw_port_dscp_map_get() was taken care only for BCM_PRIO_DROP_FIRST, not taken care for other CNG values. This has been resolved.
SDK-56043		88660_A0	During warm-boot validation, multiple issues have been found: 1. The bcm_l2_init was considered as a separate API: when called, the L2 module was detached and re-attached. This is fixed since L2 is initialized during BCM init and cannot be considered as separate API 2. When the device is initialized in TM (Traffic Management) mode, some init code was accessing by mistake uninitialized SW DB. This is fixed. 3. In L2 module, the freeze state (e.g. set by the bcm_l2_addr_freeze API) was not restored correctly after Warm-boot. This is fixed
SDK-56045	766017	56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0	Committed Information Rate (CIR) and Committed Burst Size (CBS) configured in ICAP policer were not recovered correctly during warm boot on TR3 device. The ICAP policer recovery logic is updated to fix this issue.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-56047	761668	56850_A2	The customer requested configuration of RTAG7_HASH_CONTROL_4.VXLAN_PAYLOAD_HASH_SELECT_A/B to meet their hash requirement. For Trident2 and subsequent XGS devices, 2 switch controls bcmSwitchHashVxlanPayloadSelect0 and bcmSwitchHashVxlanPayloadSelect1 have been provided to support the requirement.
SDK-56053		88650_B1 88660_A0	IP Tunnel CINT: In a GRE termination example in cint_ip_tunnel_term.c, a tunnel configuration was changed to use the correct GRE enum type.
SDK-56058	766252	56850_A2	Fixed specific sequence of (SIP, multi-DIP)-add followed by delete and then add of vxlan_tunnel_initiators.
SDK-56068	765431	56640_A0 56640_A1 56640_B0	In the previous release there was an issue reported where TR3 semlock was out of order when creating 2 OAM sessions with same vlan/port, different level. This issue of memory locks not being released in failure case in OAM code for OAM_OPCODE_CONTROL_PROFILEm and ING_SERVICE_PRI_MAPm has been fixed.
SDK-56069		56340_A0	while merging the ranges of TCP and UDP, on range not equal pointer index should increment. In this case no increment is done which leads to infinite loop on the same pointer. Fixed the indexing increment on no range match.
SDK-56071		88650_B0 88650_B1 88660_A0	OAM: For UP-MEPs, all OAM frames trapped to the FPGA/CPU will be prepended with one set of internal headers, specifically an FTMH, PPH and a FHEI, with the OAM-ID on the FHEI. Formerly some frames included two sets of internal headers.
SDK-56074	750523	56440_A0 56440_A1 56440_B0	Issue: After setting spn_BCM5644X_CONFIG to 1 to split HG2 and HG3 into GE24 - GE31, the number of priority groups for these ports were not updated. Fix: After setting spn_BCM5644X_CONFIG to 1 to split HG2 and HG3 into GE24 - GE31, the number of priority groups for these ports are changed from 7 to 0, before configuring the priority group realted registers/tables in BCM5644x devices.
SDK-56100	751146	56450_B0 56450_A0	Support has been added for APIs bcm_port_timesync_config_set() and bcm_port_timesync_config_get() for BCM5645x devices.
SDK-56108	762032	88660_A0	OAM: Enable creating accelerated MPLS OAM endpoint after Ethernet endpoint
SDK-56122	763713	All	Added PORT_INIT check to all bcm_port_XXX functions to avoid their invocation before port subsytem is initialized.
SDK-56123	753886	56243_B0 56240_B0 56243_A0 56242_A0 56242_B0	Enabled OAM endpoint addition and deletion multiple times without any error

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56128		56840_A0 56850_A2	In earlier releases, mac_xl_egress_queue_drain() blindly adds the PORT back to EPC_LINK_BMAP regardless previous EPC_LINK_BMAP state. This has been resolved.
SDK-56140	766375	56640_A1	Problem: When external TCAM table size is configured for IPv4 routes, IPv6 routes were not getting properly programmed/getting hit in the internal TCAM. Solution: Added support for this specific case of having all IPv4 routes on external TCAM and all IPv6 routes on internal TCAM.
SDK-56142	765705	56850_A0 56850_A2	In the previous release, customer reported that the rate is not accurate after changing rate from VERY HIGH PPS to low PPS. This issue had been solved by adding condicision while in _bcm_trx_rate_meter_portmode_set(), while adding dlf value, not need to refer to previous setting in register/memory.
SDK-56154		56640_A0 56544_A0 56542_A0 56541_A0 56540_A0 56524_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0 56524_B0 56540_B0 56541_B0 56544_B0 56542_B0	Enhancement:- There are 16 FP physical tcam slices with 512 entries per slice. There are 8 physical FP meter pools with 1024 entries per meter pool. Currently only 8 physical tcam slices are allowed to access the 8 FP meter pools. Requirement was to ensure the 16 physical tcam slices are allowed to access the 8 FP meter pools Support :- The 8 FP Physical meter pools are split into 16 logical meter pools so that the 16 FP tcam slices can attach to the 16 Logical FP meter pools.
SDK-56160	766445	56850_A0	In previous releases, L2 polling thread can process a MAC address insert/delete/move within a bucket, but it cannot process the scenario that a MAC address move from a bucket in a bank to another bucket in another bank. In this release, processing the scenario that a MAC address move from a bucket in a bank to another bucket in another bank has been added in L2 polling thread.
SDK-56189		88650_B0 88660_A0	Required changes in SDK in order to support KBP-SDK 1.2.3 and higher. The changes include configuration of a newly used instruction and its transport layer implementation.
SDK-56190	767623	56850_A0 56850_A1 56850_A2	In previous release, bcm_l3_route_add API may returned Not_Found if with an IPv6 VRF_GLOBAL route entry in ALPM mode even if ALPM memory table had enough space. In this release, it can be added successfully.
SDK-56193		88650_A0 88650_B0 88650_B1 88660_A0	COSQ: CNM profile can be allocated by calling bcm_cosq_qcn_config_set. Fixed an error of CNM profile initialization. Corrected the number of entries be added to template init ID.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56194		88650_A0	In Warmboot module, asserts were recently inserted to avoid modifying an uninitialized SW database. These asserts are replaced by regular error mechanism. Thus, most set/get SW DB functions are modified to return also an error value.
SDK-56195		56850_A0 56850_A1 56850_A2	In the previous release, PORT table LOCK would not be released if gport validation failed when operating PORT table. In this release, PORT table LOCK will be successfully released if gport validation fails when operating PORT table.
SDK-56199		88660_A0	OAM: when calling bcm_oam_loss_get() the near/far fields returned were mixed up.
SDK-56203		88650_A0 88650_B0 88650_B1 88660_A0	XLPORT Overrun/Underrun Workaround ---: The Arad driver implements a sequence to recognize and recover the port from XLPORT Overrun/ Underrun issue (see BCM88650 errata sheet). To activate the sequence during device init use the following soc property: custom_feature_nif_recovery_enable=1 (default is disabled on 6.3.x, and enabled on 6.4.x). The sequence might perform several iterations when trying to recover the port. To limit number of iteration use the following SoC property: custom_feature_nif_recovery_iter (default is 3). Note that from lab experience the port is recover within single iteration. Limitations: 1. The SW WA works for XLP0 only. 2. The SW WA is called during init and isnt available for dynamic port.
SDK-56215	754083	All 56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56640_A0 56850_A0 56843_B0 56841_A3 56846_A1 56841_B0 56640_A1 56640_B0 56850_A1 56850_A2	In the previous release, bcm_cosq_port_mapping_set and bcm_cosq_mapping_set returned BCM_E_RESOURCE incorrectly when there was one unused profile of the COS_MAP table on Trident/Trident2/Triumph3. In this release, this issue has been addressed by setting the MC_COS1f and UC_COS1f of the COS_MAP table at the same time.
SDK-56222	767209	56846_A0	During warm boot upgrade from SDK 6.2.9 to SDK 6.3.3, data qualifiers (UDFs) are not recovered in field module as there is a mismatch between field qualifier count (bcmFieldQualifyCount) in 6.2.9 and 6.3.3. The more field qualifiers are added in SDK 6.3.3. The issue is fixed in SDK 6.3.8 by storing bcmFieldQualifyCount in scache and by mapping the recovered field qualifier Id to the appropriate data qualifier.
SDK-56225	767847	88650_A0	E2E scheduler port shaper is limited from below. Added fix such that in case requested rate is lower than allowed, the minimal rate will be set.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56244	765693	`56840_A0 56850_A2`	The guideline for `bcm_cosq_gport_mapping_set` is improved in this release by specifying that it can be used on chips which support ETS(Enhanced Transmission Selection) feature regardless of that the ETS mode is enabled.
SDK-56249		`88650_B0`	Egress shaper for ILKN interfaces don't work properly, causing unexpected behavior(wrong rates). The shaping for ILKN interface is set using: `bcm_cosq_gport_handle_get(0,bcmCosqGportTypeLocalPort,gport_info);` Fixed!
SDK-56253 PHY-1417	768344	`84328_B0`	Issue Reported: G40 Port Disable not working as expected Fix: Register sequence is modified to fix this issue.
SDK-56254	765972	`88650_A0 88650_B0 88650_B1`	OAM: when creating a MIP and calling `bcm_oam_action_set()` for that MIP, the profile was handled incorrectly and might have caused packet drop. This issue was fixed and resources are properly managed and freed.
SDK-56272	767442	`56340_A0`	In earlier releases, Helix 4 GS GE48 was using incorrect lane information in the PHY driver. The port affected using quad smgii was using lane 2 and not lane 0, This has been fixed.
SDK-56280		All	BMW CPU platform was removed from SW and SQA nightly builds and tests 2 years ago. BMW is not built or validated on any recent SDK releases in either SDK-6.3.x or SDK-6.4.x train. The last SW release that supported BMW was SDK-6.2.0 on Aug 16, 2012. BMW binaries were removed starting with SDK-6.2.1 (Oct 22, 2012) onward. By the time of SDK-6.4.2 release, it will have been exactly 2 years since the platform was discontinued in official releases. The problem is that SDK Platform guide has not been updated in the longest time. Now that we are about to publish an updated version together with SDK-6.4.1 release, this JIRA will update the document to match what has been published on DocSAFE. Releases posted on DocSAFE did not include BMW binaries for 2 years now.
SDK-56291	768458	All	The definitions of `COUNTER_ATOMIC_BEGIN/END` in COUNTER thread adopted `sal_splhi/sal_spl` as mutex lock to protect some small critical sections, which can cause a considerable performance loss due to its overhead and coverage scope. Replacing the old one with a new lock mechanism, the `sal_spinlock` primitives can be more efficient especially for protecting small critical sections somewhere like in COUNTER thread. `sal_spinlock` can be used in Linux user space, Linux kernel and vxworks, even in interrupt context. To be noted, it can't be used recursively.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56295		88650_A0 88650_B0 88660_A0	BFD accelerated endpoint that is handled in remote gport - SW DB is not restored correctly after WB.
SDK-56306	769032	88030_A0	note.
SDK-56317		56846_A0 56846_A1	In previous releases, created multipaths more than max capacity could corrupt existing ECMP groups and return wrong value -1 if ECMP group size of TD+ configured to 256 as TD device. In this release, it returns BCM_E_FULL(-6) if creating ECMP multipaths more than max capacity.
SDK-56340	755455	88650_A0	Port enable sequence was fixed to support 1588 on 1G ports.
SDK-56350		88650_A0 88650_B0 88650_B1 88660_A0	The "multiple packet dequeue" feature which is meant for usage in low latency credit request profiles can now be configured using the bcm_cosq_delay_tolerance_level_set/get APIs. The feature is activated for a credit request profile if the following new flag is used in the flags field of the structure: BCM_COSQ_DELAY_TOLERANCE_IS_LOW_LATENCY . In release 6.4.1 all the credit request profiles named BCM_COSQ_DELAY_TOLERANCE_*_LOW_DELAY will have this feature set. In 6.3.* releases the default profiles are not changed, though this can be done manually. Example of changing one predefined profile manually: bcm_cosq_delay_tolerance_level_get(unit, BCM_COSQ_DELAY_TOLERANCE_200G_LOW_DELAY, &delay_tolerance); delay_tolerance.flags |= BCM_COSQ_DELAY_TOLERANCE_IS_LOW_LATENCY; bcm_cosq_delay_tolerance_level_set(unit, BCM_COSQ_DELAY_TOLERANCE_200G_LOW_DELAY
SDK-56352		88660_A0	Fixed ECN (Explicit Congestion Notification) to work correctly in 88660
SDK-56353 SDK-56332	768573	88650_A0 88650_B0 88660_A0	In Policer rate computation function, the exponent and mantissa configuration was fixed in case the required value is too small. When allocating a meter with a very low rate (for instance when using bcm_policer_config_t.max_pkbits_sec = 128), the driver produces an error, even though this is a valid rate. This is now fixed.
SDK-56355	767767	88660_A0	In L2 module, when working in centralized mode, the LIF-valid bit entry was not received correctly on learn events (i.e., when the CPU was inserting learnt entries via BCM SDK). The LIF-valid bit is now set correctly on the learn events and matches the payload of the device learned entry.
SDK-56379		All	Support has been added for resolving the modern GPORT types (TRILL, VXLAN, NIV, L2GRE, etc.) in the Diag Shell.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56387	769040	56450_B0 56450_A0	Support has been added for IFP qualifier bcmFieldQualifyRouterAlertLabelValid for BCM5645x devices.
SDK-56409		All	PTP clocks can now be re-created in order to change the "immutable" clock parameters. The only restriction is that the number of clock ports on re-creation may not exceed the value used on the initial creation.
SDK-56410	769158	88650_A0	TCDP mapping using bcm_cosq_gport_egress_map_set uses profiles of mapping, and associate each port to relevant profile. The API supported up to 4 different profiles, although HW support up to 8 profiles (when new profile is required but not avaliable, the API return an error). The API was fixed to support 8 profiles as the HW.
SDK-56425	767797	88650_A0 88660_A0	SER interrupts were not signaled to CPU and not counted, due to being masked by a set of override bits called monitor bits. This was fixed to allow proper logging and handling of SER interrupt events by the SDK.
SDK-56439		88650_A0 88650_B0 88660_A0	Ethernet OAM does not recover from Warm-boot. This is fixed.
SDK-56440		88650_A0 88660_A0	MPLS Tunnel initiator clear all API does not clean up MPLS WB information as well.
SDK-56441		88650_A0 88660_A0	During Warmboot in vswitch module, the VSI MSTP was always restored, even if it was cleared before the warmboot. This restoration is skipped upon Warmboot.
SDK-56446	759287	88650_A0 88660_A0	Fix low_vid verify value in bcm_vswitch_port_delete function (arad_pp_frwrd_trill.c).
SDK-56447	763576	88650_A0 88660_A0	When creating an ECMP group using bcm_l3_egress_ecmp_create, if the 'ecmp' parameter is NULL, a segmentation fault was occurring. This is now fixed - the software checks that the 'ecmp' parameter is not NULL.
SDK-56451		88650_B0 88660_A0	Required changes in SDK in order to support KBP-SDK 1.2.3 for external TCAM are introduced.
SDK-56452	760578	56450_B0 56450_A0	When 1 + 1 protection switching is enabled/ disabled (with label swapping on IPMC group), the MPLS::LABEL_ACTION_SWAP field of EGR_L3_NEXT_HOP table need to be set/ cleared respectively to achieve the functionality. This support has now been added.
SDK-56455 SDK-56327	769233	56224_B0 56224_A0	Issue :- IpType Qualifier was not recovered properly after warmboot. Fix :- Recovery of IpType Qualifier was not handled properly in BCM56624. Added Code to recover IpType qualifier after warmboot.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56464	765386	56640_A0 56640_A1 56640_B0	Problem: When there is no signature configured, if there is traffic being sent to signature matching engine(SME) for deep packet inspection, the SME goes into a hung state. Even after configuring some signatures later on, it doesn't indicate any match though it receives matching traffic. Solution: Do not let any traffic to be forwarded to SME until at least one signature is configured successfully. Also, stop the traffic from being forwarded to SME while detaching the last active engine. This is achieved by modifying flow tracker configuration register field.
SDK-56476		88650_A0 88660_A0	In Field Processor's diagnostics, in case field groups were created however no entry was inserted, an error is produced when calculating the number of entries from an empty bitmap. This is fixed.
SDK-56482	768774	56450_B0 56450_A0	Added support for associating a MPLS label to a given protection switching group for BCM5645x devices. API bcm_mpls_tunnel_switch_add() can be used to achieve this by passing the protection switching group id in "failover_id" member of structure "bcm_mpls_tunnel_switch_t".
SDK-56492	769633	56850_A0 56850_A1 56850_A2	The related EGR_PORT_TO_NHI_MAPPING was not cleaned when the last port was removed from the trunk where a VXLAN logical port is created. Now it is fixed by adding the specific implementation for VXLAN.
SDK-56495	768732	88650_A0 88650_B0 88650_B1 88660_A0	In Field Processor, at Egress, the support of two new qualifiers is introduced: bcmFieldQualifyISid (MAC-in-MAC I-SID) and bcmFieldQualifyMplsForwardingLabelAction. Both qualifiers are mapped internally to the EEI value.
SDK-56507		56640_A0 56640_A1 56640_B0	Previously bcm_cosq_port_bandwidth_set() failed on 56640 HSP ports. HSP ports handling for bandwidth_set API has now been added.
SDK-56514		56850_A0 56854_B0	In previous releases, SER correction for MMU CTR block was not implemented. In this release, MMU CTR block SER correction logic has been implemented. Once parity error is detected in tables in MMU CTR block, the corrupted table entry will be cleared.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56522		56640_A0 56850_A0	Issue here is SDK support is missing to recover EFP Secondary selectors during Warm Boot. So EFP qualifiers which need secondary selectors will fail to be recovered in Warm Boot. Now support has been added for recovering all EFP secondary selectors come from following different registers 1. EFP_CLASSID_SELECTORr - (HX4, KT2, TR3, TD2) 2. EFP_KEY4_DVP_SELECTORr - (HX4, KT2, TR3, TD2) 3. EFP_KEY4_MDL_SELECTORr - (HX4, KT2, TR3, TD2, GreyHound, Enduro) So this fix is applicable to HX4, KT2, TR3, TD2, GreyHound, Enduro
SDK-56533	769718	56850_A2	Fixed multicast module to return error when deleting member from a MC group that was already destroyed.
SDK-56554	770975	56850_A0 56850_A1 56850_A2	Support has been added for the ability to transmit even if port is down.
SDK-56572	771276	88660_A0	When using external TCAM for forwarding, serial IP and RPF, then high rate lookups return sometimes wrong results. This is fixed: serial lookups in external TCAM are always returning reliable results at any supported rate.
SDK-56577		88650_A0 88660_A0	Removing sw database MC-ID -> nickname. Nickname can be extracted from trill_port_id database (encap_id field).
SDK-56578		88650_A0 88660_A0	New sequence for ECMP creation using forward-group port instead of trill-port-ecmp.
SDK-56580	772058	88650_B1 88660_A0	QOS: Fixed the ability to set Inner-PCP to TC/DP table in bcm_qos_map_add.
SDK-56581		88650_A0	In Field Processor diagnostics, the actions offsets are incorrect when cascaded action is used. This is fixed.
SDK-56591	768899	56850_A0 56850_A1 56850_A2	New API bcm_l3_egress_stat_counter_sync_get() added to retrieve l3 egress stats after updating the software copy of the counter value with the hardware counter value.
SDK-56594 SDK-57957	769099	56440_A0 56445_A0 56440_A1 56445_A1 56444_A1 56450_A0 56340_A0 56449_B0 56445_B0 56440_B0 56447_B0 56443_B0 56441_B0 56446_B0 56448_B0 56344_A0 56342_A0 56442_B0 56342M_A0 56340M_A0 56455_A0 56456_A0 56450_B0	In earlier releases scheduler configuration with weight value 0 was incorrectly configured in WRR mode. This has been fixed in this release to configure scheduler configuration with weight value 0 to be in STRICT PRIORITY mode.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-56597	772109	56850_A2	soc_alpm_insert: Route Insertion Failed due to DEFIP AUX Operation timeout. On expiry of poll for ALPM hardware operations, soc_timeout_check requires that the status register needs to be read one more time to confirm operation has completed. This support has been added.
SDK-56607		88650_B0	Fix initial shaper to interface mapping. This fix has no functional impact.
SDK-56608	765207	56450_A0 56450_B0	When the physical port associated with MPLS port is replaced by using bcm_mpls_port_add() API with flag BCM_MPLS_PORT_REPLACE, the properties associated with old physical port is not cleared. Appropriate check has been added to clear the properties associated with the old physical port for BCM5645X devices.
SDK-56610	772885	56450_A0 56450_B0	gport_attach function can be called passing cosq value as 0,1,2.. so on. When coaq value was passed instead of -1, code was not handling it properly, So same hw_index was allocated again and again, Now checks are provided so that unique hw_cosq value is assigned for different values of cos
SDK-56611	772970	88650_A0 88650_B0 88660_A0	After Hard_Reset was called , CPU port was stuck. Resolved in the hard reset code by resetting CMIC TXi credits.
SDK-56615	772971	56450_A0 56450_B0	WRR scheduling under sub ports could not work due to missing weight configuration in L0 nodes. This configuration issue has been corrected to get the expected scheduling behavior.
SDK-56628		88660_A0	BFD: for BFD endpoints of type bcmBFDTunnelTypeMpls (BFD PDUs are encapsulated by UDP, IP, MPLS, Eth), IP TOS, TTL may be configurable through the fields ip_tos, ip_ttl. Note that the protocol dictates that the IP TTL be set to 1.
SDK-56629		88650_A0 88650_B1 88660_A0	When compiling with INCLUDE_KBP compilation flag, a large memory allocation for Field Processor software state was performed, related to external TCAM. This large memory allocation is now performed only if ELK usage is indicated via SOC properties.
SDK-56635		88650_A0 88650_B0 88660_A0	In some scenarios, trunk ports lb_key_min and lb_key_max values do not cover all lb_key range [0:255] which results in packet drop. This issue is fixed.
SDK-56636		88650_A0	At SOC layer, a new mechanism to improve the performance of entry insertion for Large-Exact-Match, Small-Exact-Match and TCAM databases has been implemented. By default, this mechanism is enabled. To disable this mechanism, unset the compilation flag ARAD_FAST_REGISTERS_AND_FIELDS_ACCESS.
SDK-56641		56850_A2	In earlier releases, VxLAN multicast was treated as non-Layer3 multicast. It caused VxLAN multicast group to still have members after being re-created. This has been resolved.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56644		56440_B0	EGR_L3_NEXT_HOP table has overlapping views, for ex: L3, MPLS, SD_TAG etc., and for a given entry in EGR_L3_NEXT_HOP table the fields corresponding to a particular view, decided by ENTRY_TYPE field, should only be modified. But some fields of L3 view (overlapping with MAC_DA_PROFILE_INDEX field of MPLS view) were always getting modified resulting in wrong EGR_MAC_DA_PROFILE entry getting overwritten when a given L3 egress object is updated using the flags BCM_L3_REPLACE | BCM_L3_WITH_ID. Protection (i.e, check for appropriate ENTRY_TYPE value before modifying fields in L3 view) has been added to overcome the issue.
SDK-56646		88650_A0 88660_A0 88670_A0	Fixed a problem in bcm_mpls_port_add. The issue caused the driver to crash with a segmentation fault when the API is called with the REPLACE flag.
SDK-56647		88650_A0 88650_B0 88660_A0	In FCoE, when adding a route via bcm_fcoe_route_add API with flags BCM_FCOE_LOCAL_ADDRESS | BCM_FCOE_HOST_ROUTE, the entry was not be added correctly to the forwarding database. This is fixed.
SDK-56649	772044	88660_A0	In metering when the SOC property policer_color_resolution_mode is set to 1, the meter processor outputs the following DP values: green - 0 yellow - 1 meter processor red - 2 ethernet policer red - 3. Due to a software bug, when the meter processor gave a packet a color of yellow, the actual DP would be 2, instead of 1. This is now fixed.
SDK-56657 SDK-54730		88660_A0	Currently, unless specified by SOC property, ethernet policers drop all packets that arrive red to the device. Color blind ethernet policers allow to do rate policing even for packets that arrive red to the device. This fix introduces the ability to change ethernet policers to be color blind or color aware dynamically. To set color blind ethernet policing, both the ethernet policer and aggregate policer associated with a port and traffic class must be set to be color blind. To set an ethernet policer to be color blind, the BCM_RATE_COLOR_BLIND flag can be used when calling bcm_rate_bandwidth_set. To set an aggregate policer to be color blind, the BCM_POLICER_COLOR_BLIND flag can be used when calling bcm_policer_set with an aggregate policer.
SDK-56688		56340_A0	In the previous release the packet/byte fields were not working correctly in regex reports. The packet and Byte counter registry values are now retrieved and updated in the match reports.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-56691		88650_A0 88650_B0	In Rx parsing, the src_gport and dst_gport meaning were unclear. It is fixed to: - src_gport: incoming port where the packet was introduced into Arad - dst_gport: outgoing port where the packet exits from Arad
SDK-56693		56340_A0	When only engine 0 is enable, the CSF table will not be updated no toggling on CSF valid signal. When other engine are enabled, the CSF table will be loaded when the valid signal of other engines toggle. Hence enabling starts from engine 1 and engine 0 will be enabled at last.
SDK-56700	774184	88650_A0 88650_B0 88650_B1 88660_A0 88670_A0	When calling bcm_mpls_port_add with pwe id > 32K, error printouts are provided but the API returns BCM_E_NONE. This is fixed and error is returned.
SDK-56701	773800	All	In earlier releases diag shell would intermittently crash in "l3 egress show" command. This has been resolved.
SDK-56709	773764	56334_B0 56334_A0	Issue: ==== Remote trunk identifier bit has to be ignored while setting the srcTrunk mask. The MSB of the modId represents Remote trunk bit and hence it was calculated based on the width of the qualifier. However, the width of the qualifier varies for different devices. Due to this, for devices like Enduro, the bit was positioned wrongly and was ignoring trunk bit instead of the remote trunk bit. Fix: === Instead of using the width of qualifier, the bit position of the trunkBit minus 1 (trunk_bit_pos - 1) is used to calculate the remote trunk bit position and ignoring the bit by masking the bit to 0.
SDK-56714	758491	56450_A0 56440_B0	Issue : --------------- In katana and katana2 the rqe_port_config register was programmed with cos_mode=1 when extended queueing was enabled but cos_mode =1 is not valid for this register and this causes traffic to go through cos 0 always Fix ----- If extended queuing is enabled then we program cos_mode with value 0 in RQE_PORT_CONFIG .
SDK-56720	769698	56224_B0 56224_A0	Output of "trunk show" command displays the port names correctly by resolving the gports for all XGS devices.
SDK-56725		56850_A0 56855_A0 56850_A2	In previous release, the functions bcm_vxlan_stat_attach and bcm_vxlan_stat_counter_get took high execution time, about 13000 usec per call, which couldn't meet customer expectations. In this release, the functions have been optimized, and they takes about 100 usec per call. The performance has been improved.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-56741	763657	56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56340_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0 56344_A0 56342_A0 56342M_A0 56340M_A0	In earlier releases, the validation on the PORT for ETS was incorrect on the return value which might lead to wrong COSQ mapping. Fixed the validation.
SDK-56753		56640_A0 56643_A0 56640_A1 56643_A1 56640_B0 56643_B0	Problem: Due to a hardware bug, the hardware team had recommended to disable bus parity protection for a bunch of memories which includes IESMIF. However, the SDK still has the bus parity enabled on IESMIF and this is causing spurious parity errors in the cases where ESM accesses are involved. Solution: Disable the bus parity protection for IESMIF by default, to workaround the hardware issue.
SDK-56756	773877	56540_A0 56540_B0	Previously, "l3 ip6route show" command was broken on Firebolt-4. This is due to that soc_feature_l3_shared_defip_table is not supported on Firebolt-4 and thus bcm_switch_object_count_get called in this command returns an error. It is fixed by adding the additional check on soc_feature_l3_shared_defip_table to avoid calling bcm_switch_object_count_get for Firebolt-4.
SDK-56761		56540_A0 56340_A0 56540_B0	In Apollo2 and Helix4 devices, during an OAM CCM timeout event, remote endpoint index passed from SDK to OAM event callback function was not correct, this issue has been addressed.
SDK-56763	772471	56850_A0 56850_A1 56850_A2	In the previous release, the API bcm_cosq_gport_bandwidth_set would set the shaper on a wrong scheduler node. In this release, this issue has been addressed by setting the software resources which have been assigned to the HSP ports.
SDK-56765		88660_A0	Add driver support to new Arad SKU - 88363
SDK-56770	774767	88650_B0	Trill learning: In TRILL multicast, ingress learning, MACT learning is disabled at Egress Router-Bridge for TRILL multicast packets otherwise unrelated MACs (Link Layer SA) are learned.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-56779	774862	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In earlier release, TD2 had 48 HIGIG trunks and SDK was not able to record/maintain the bitmap of higig trunk override id which was larger than 31. This issue has been fixed in this release.
SDK-56781	774909	56850_A0 56850_A1 56850_A2	The root cause of this issue is that when l2 addresses are learnt on Y pipeline, the hit bits of the corresponding L2X table entries are not set. But `bcm_l2_matched_traverse` function will read all L2X table entries and check the hit bits, so the traverse function can't find the l2 address learnt on Y pipeline. Code has been added to update the hit bits of L2X entries when the L2 addresses are learnt on Y pipe line.
SDK-56786	773228	All	Support has been added for displaying counter register's alias name which register's name larger than 13 characters
SDK-56789		56340_A0 56547_A0	For 802.1AS packet, RX and TX timestamps are now enabled for BCM56340 family of devices
SDK-56801	774468	56440_A0 56445_A0 56440_A1 56450_A0 56440_B0 56450_B0	In earlier releases, Enabling of `tcam_protect_write` resulted in incorrectcomputation of the number of entries per slice on Katana. This issue has been fixed by correcting size of the FP TCAM value used for computing the size of each slice.
SDK-56805		88660_A0	ARP downstream checking didn't use separately, now the fixes resolve this issue.
SDK-56821		56820_B0	In Scorpion, IP Packets with 0x9100 (other than default TPID 0x8100) outer tag are treated as untagged and non IP packets when these packets ingress on YPIPE and egress on XPIPE. This behavior is detected by the EFP when it is configured to match anything beyond the L2 header. The software work around is added in SDK to fix this issue by changing the access type of the per-port register `EGR_SRC_PORT`.
SDK-56840	772939	88650_A0 88650_B0 88650_B1 88660_A0	In MAC-in-MAC, when using API `bcm_l2_addr_add(),` multicast group destination was not supported in BMACT Forwarding table. Multicast group destination is now supported and can be added to BMACT forwarding table.
SDK-56848	776418	82328_A0	Added PHY BCM82322 support. This PHY supports 10G,20G and 40G modes
SDK-56850	776440	56450_A0 56450_B0	Issue : Support for ECAP CopytoCpu is missing on KT2. Fix : Added Support for ECAP CopytoCPU in KT2 in SDK
SDK-56854		88650_A0 88650_B0 88660_A0	In FCoE zoning, when adding an entry, all entry actions were applicable (allow, deny, redirect), but the same action (allow) was always executed. A validation is introduced so that only the action allow is applicable.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56869		56450_A0 56450_B0 56640_B0 56440_B0	PTP master sends Announce messages to PTP slaves. PTP slaves compares announce messages received from several PTP master to decide which PTP master to choose to synchronize time. Following three new fields are added to PTP master information structure. ClockAccuracy: The clockAccuracy indicates the expected accuracy of a clock when it becomes grandmaster or in event it becomes grandmaster. Various granularities are possible. This specifies time is accurate to within 25 ns/100 ns/250 ns/1 ?s/2.5 ?s/10 ?s etc. OffsetScaledLogVariance: The offsetScaledLogVariance indicates inherent precision of a clock. This is the precision of the timestamps included in message issued by clock when it is not synchronzied to another clock using the protocol. The reference clock when not synchronized to another clock may be an atomic clock, a GPS receiver, a stable local oscillator, a suite of clocks synchronized via NTP, etc. These sources may contribute to the variance estimate. The value of offsetScaledLogVariance can also be a staic constant determined by manufacturer. StepsRemoved: The distance measured by the number of boundary clocks between the local clock and the foreign master is used when two Announce messages reflect the same foreign master. The distance is indicated in the stepsRemoved field of Announce messages.
SDK-56876	776002	56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0	The scheduler configuration with weights value 0 will be considered as STRICT_PRIORITY. Fixed the same behavior in SDK.
SDK-56878	776733	88650_B1	The bcm_cosq_control_set/get(unit, 0, 0, bcmCosqControlAdmissionTestProfileA, bitmap) APIs did not work correctly if the bitmap bits for PFC or LLFC VSQ types were set. This was fixed.
SDK-56884		88650_A0 88650_B0	MIM: DEFAULT BEHAVIOR CHANGE . Encoding of returned handler station_id for MIM is now changed in l2 station APIs. bcm_l2_station_get() API failed in some cases when LSB for MyMac was considered to be global instead of per ingress port. This happened when the MIM global LSB bit in the created station_id was wrongfully set due to an overlap in the station_id encoding. This is fixed by changing the encoding of the station_id so that there is no overlap with the MIM LSB global indication bit. The MIM global LSB indication bit in station_id changed from bit 7 to bit 16.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56887		88660_A0	Default value of Chicken bit EGQ_CFG_BUG_FIX_CHICKEN_BITS_REG_1 CFG_BUG_FIX_87_DISABLE was changed to disable (instead of enable) as it doesn't provide any new functionality.
SDK-56888 SDK-56945	742236	88650_A0 88650_B0 88650_B1 88660_A0	Support reflector functionality in accordance with RFC-2544 (benchmarking methodology). This JIRA contains IP+MAC swap functionality (swap the SIP with the DIP, SA with DA) as well as a light MAC-only-swap functionality (swap the SA with the DA). For the former, the soc property RFC2544_reflector_mac_and_ip_swap_port should be set to the reflector port. All packets arriving at the ETPP with the Out-TM-port set to the reflector port will have their MAC addresses and IP addresses swapped, and the packet will be prepended with a PTCH with the SSP set to the original Out-PP-Port. The reflector port should be defined as a recycle port and the IP routing should be done at the second pass. The light MAC-only swap functionality can be used analogously with the soc property RFC2544_reflector_mac_swap_port For a more detailed account (For example setting an egress-PMF rule modifying the Out-TM-port), refer to cint_benchmarking_methodology.c
SDK-56903		56850_A0	Adding a flag BCM_NIV_VNTAG_L_BIT_FORCE_1 to choose if frames can be headed back towards the Interface Virtualizer that it originated from.
SDK-56913	759274	All	In earlier releases on overflow the DMA timeout/overflow stat was cleared only when the entries were available. On entry empty this was not getting cleared. Fixed in the changes when the entry is empty.
SDK-56917	777278	56340_A0	SMEmatchnotreportedfortwitterandwebexsignatures due to hex representation of ASCII. Provided support in SDK API bcm_regex_match_set() to parse hex representation of ASCII Alphabets.
SDK-56925		88650_A0 88650_B0 88650_B1 88660_A0	PON: In previous release, DHCP IPv6 anti-spoofing wasn't working when soc property l3_source_bind_mode is IPV6, now fixed this issue.
SDK-56929		56850_A0 56850_A1 56850_A2	In earlier releases, next hop information was not initialized before using it. This has been resolved.
SDK-56931		56850_A0 56850_A1 56850_A2	In previous releases, the API bcm_l3_egress_get returned BCM_E_INTERNAL in vxlan case. A new case _bcmVpTypeVxlan has been added to fix this issue. Now if the case is vxlan, the egr->port will be set to vxlan and the API will return BCM_E_NONE.
SDK-56954		56850_A0	In earlier releases, source trunk table was not being cleared up if a customer used an incorrect sequence. This has been resolved.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-56956	774358	88650_A0 88650_B0 88660_A0	In Field Processor, in Direct Extraction field groups, it is possible to set a bias value as part of the extraction field configuration. The procedure failed when the bias value is negative. This is fixed.
SDK-56960		88650_A0 88650ACP_A0 88650_B0 88650_B1	QOS: Qos map id can be destroyed by calling bcm_qos_map_destroy. Improvement in entry deletion for bcm_qos_map_destroy by adding new SW DB to record each entry is occupy or not.
SDK-56961		88660_A0	BFD: When calling bcm_bfd_endpoint_create() with the flag BCM_BFD_ENDPOINT_REPLACE set and type==bcmBFDTunnelTypeMplsTpCc, static registers were mismanaged, causing such calls to fail.
SDK-56962	776131	88650_A0 88650_B0 88660_A0	The OAM DM DOWN program at the egress editor has been fixed so that only appropriate packets will select this program. Previously this program was catching other packets as well which cased outgoing packet corruption.
SDK-56964		56850_A1 56850_A2 56850_A0	In earlier releases the related EGR_PORT_TO_NHI_MAPPING was not cleaned when the last port was removed from the trunk where a VXLAN logical port was created. This is fixed by adding the specific implementation for VXLAN.
SDK-56975	774350	56850_A0 56850_A1 56850_A2	Customers requested more granularity in bcm_vxlan_vpn_create. To enable this modifications were made to BCM_VXLAN_VPN_WITH_VPNID to meet this goal. Before this change, when customer created a vpn, BCM_VXLAN_VPN_WITH_VPNID us required, and both VFI and VNID were created.After this change, the behavior is as follows: When create a VXLAN VPN: If use BCM_VXLAN_VPN_WITH_VPNID, both VFI and VNID will be created. If not. use flag BCM_VXLAN_VPN_WITH_VPNID, only VFI will be created. When updating an existing VXLAN VPN(BCM_VXLAN_VPN_REPLACE should be used. If use both BCM_VXLAN_VPN_REPLACE and BCM_VXLAN_VPN_WITH_VPNID, both VFI and VNID will be created. If only use BCM_VXLAN_VPN_REPLACE, the VNID will be removed.
SDK-56980	777710	56240_B0	In previous releases, If the given port was configured with WRR scheduling and then warmboot was done the SW did not recover the correct scheduling algorithm back after the warmboot .The hardware continued to have correct value . This has been resolved.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-56988		`56850_A0 56850_A1 56850_A2`	Customer wanted to use VLAN and VFI flex counter simultaneously. But in the previous release, SDK assigned VLAN and VFI counter in the same pool, and this would cause the VFI counter to not be updated when the packet hit two memories. Now the customer can use the SOC property `ing_share_flex_counter_pool=split(vlan,vfi)` to prevent VLAN and VFI from sharing the same pool.
SDK-56991	778526	`56850_A2`	In earlier releases, when using `bcm_vxlan_port_add()` API with `BCM_VXLAN_PORT_REPLACE` flag, it will clear the flex counter configuration if this vxlan port has attached with flex counter. This has been resolved.
SDK-56994		`56850_A0 56850_A1 56850_A2`	It was found that network facing flex counters were not working for both bcmStatGroupModeSvpType and bcmStatGroupModeSvpType group modes. After investigation we located the RCA was the counter offsets were not set correctly in previous implementation. The issue was fixed by adjusting the counter offset for both bcmStatGroupModeSvpType and bcmStatGroupModeDvpType group modes.
SDK-56995	777713	`56845_A2 56850_A0 56850_A1`	In the previous release, when using RPCs and calling `bcm_vxlan_stat_counter_get()`, the values in the `counter_indexes[]` parameter are not being properly propagated from the client to the server. This has been resolved.
SDK-56999	773690	`56850_A0 56850_A1 56850_A2`	Added support for port extender failover.
SDK-57002	778714	`56850_A2`	In earlier releases, SDK code was not able to resolve the ports for which id was larger than 64 in BITMAPf of `IFP_REDIRECTION_PROFILEm` table on TD2 after warmboot. This has been resolved.
SDK-57004		`56640_A0 56340_A0`	In previous release, the schan response type for devices with ISM, e.g. Triumph3 and Helix4 is not properly checked. The following response types `SCHAN_GEN_RESP_L2_MOD_FIFO_FULL`, `SCHAN_GEN_RESP_MAC_LIMIT_THRESHOLD` and `SCHAN_GEN_RESP_MAC_LIMIT_DELETE` have been added in schan response type checking in the routine `soc_mem_generic_insert()`.
SDK-57009		`56850_A0 56850_A1 56850_A2`	In previous releases, `bcm_vxlan_stat_detach` took high execution time because redundant memory operation was executed. In this release, we remove memory read operation and use `soc_mem_write` instead of `soc_mem_write_range` conditionally to save time, then the execution time can be reduced a lot.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57027		`56850_A0 56850_A1 56850_A2`	In earlier releases, Trunk useful information was cleared by VXLAN API.This has been resolved.
SDK-57032		`56850_A0 56850_A1 56850_A2`	In earlier releases, `bcm_vxlan_port_get()` could not get the `BCM_VXLAN_PORT_DROP` and `BCM_VXLAN_PORT_MULTICAST` flags correctly. This has been resolved.
SDK-57034	775986	`56450_A0 56242_A0`	Issue : Packet based WRED profiles was not restored properly in katana/katana2 after warmboot in earlier releases .As part of fix during warmboot we scan packet based WRED table and update the software profile.
SDK-57038		`88650_B1`	Documentation only: `stat_if_pkt_size` description in config-sand.bcm example was misleading. The correct description can be found in user manual or in property.h. Description in config-sand.bcm fixed as well.
SDK-57054	778731	`88650_B1`	add more detail prints and update the UM. Changing jira to improvment
SDK-57075		`88650_B0`	Arad initialization time significantly improved for channelized interface configuration.
SDK-57077		`88650_A0 88650_B0 88650_B1 88660_A0`	IMPORTANT CHANGE FOR PWE P2P: OAM PWE P2P was not identified as OAM in the classifier because of wrong lif id (0 value instead of the real LIF-ID value). This is fixed by setting valid LIF-ID for PWE P2P. The change may cause same-interface to be invoked for PWE P2P case when In-LIF PWE P2P ID is equals Out-LIF ID.
SDK-57078		`88650_A0 88650_B0 88650_B1 88660_A0`	OAM: Supporting down MEPs in the format CFMoEthoMplsoEth in OAM classifier. In order to enable initialization of the OAM TCAM to identify CFMoEthoMplsoEth, set soc property `custom_feature_oam_downmep_pwe_classification` to 1. This feature supports CFM identification per-md level only. This feature does not support identification per opcode. All CFM packets will be associated with opcode=1 (CCM). Inner Ethernet frames with 0 or 1 VLAN tags preceding the CFM EtherType are supported. For a more detailed explanation (including examples), consult `cint_oam_cfm_o_eth_o_pwe_o_eth.c`
SDK-57080	766661	`88650_B1`	TRILL and FCoE could not be supported simultaneously on the same device, due to an overlap in FLP (i.e. forwarding HW block) programs allocation. TRILL and FCoE can now be supported and coexist on the same device.

Table 117:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
SDK-57082		`88650_A0 88660_A0`	Important Note: the default Drop Precedence (DP) mapping of a yellow packet with DP=2 has been changed. Usually the final DP (Drop Precedence) given by the meter (or the In-DP) is unchanged, and can be from 0-3. In the past the final DP was always changed from 2 to 1 when passed to ingress, meaning that the only available DP results were 0, 1 and 3 (at ingress). To support this old behavior the SOC property `policer_color_resolution_mode` is introduced. When `policer_color_resolution_mode=1,` if the final DP is 2, this DP is mapped to 1 instead (at ingress).
SDK-57083	776583	`88650_B0 88650_B1 88660_A0`	IMPORTANT: for improved performance after `bcm_field_group_install` call, it is recommended to set `USING_TCAM_PRIO_LIST_INVERSE_SCAN` compilation flag. In Field processor entry insertion procedure, the user can: - after initialization, define all the entries and then insert them in one call `(bcm_field_group_install)` - on-the-fly, insert the entries dynamically one by one `(bcm_field_entry_install)` The advantage of the first case is the absence of TCAM shuffling, since the entries are sorted according to their priority before their insertion. In this case, the limiting factor in the entry performance was the entry insertion in the priority sorted list, an internal data structure detailing for each priority the acceptable TCAM location range. The scanning of this list was always performed from the first node to the last one, even if in the sorted case the inserted entry was the last one. This scanning has been changed to scan from the end, if the compilation flag `USING_TCAM_PRIO_LIST_INVERSE_SCAN` is set. We highly recommend to users to set this compilation flag for performance improvement.
SDK-57085		`88650_A0 88660_A0`	If `bcm_mpls_tunnel_initiator_create` is called with `WITH_ID` flag and an existing egress tunnel id, this is illegal configuration. We added a check to verify this won't happen.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57100	778739	56850_A0 56850_A1 56850_A2	In Trident2, IP_FRAG_INFO(2bit) is defined in 5 field selectors (F1_6, F1_15, F2_1, F3_3 and IFP_PAIRING_FIXED). But In SDK ,IP_FRAG_INFO in these 5 different selectors are initialized with 2 different qualifiers as below which is wrong. Modify the SDK to make it consistent i.e use bcmFieldQualifyIpFrag qualifier at all places. F1_6 - initialized for bcmFieldQualifyIpInfo F1_15 - initialized for bcmFieldQualifyIpFrag F2_1 - initialized for bcmFieldQualifyIpFrag F3_3 - initialized for bcmFieldQualifyIpInfo IFP_PAIRING_FIXED - initialized for bcmFieldQualifyIpFrag Now IP_CHECKSUM_OK is 1 bit field and part of FIXED part of IngressFieldProcessor key. Currently SDK does not have support for this 1 bit field and bcmFieldQualifyIpInfo qualifier is used to initialize IP_CHECKSUM_OK bit.
SDK-57102	779185	56850_A0 56850_A1 56850_A2	In earlier releases, If adding l3 host entry to HW failed, SDK should decrease the related reference count but this function did not work when it has a multipath flag. This has been resolved.
SDK-57104	779184	56526_A0 56524_A0 56521_A0 56526_B0 56524_B0	For BCM_5652x devices, whenever a tpid other than the default tpid was created, reference count of default tpid was decremented once but was not incremented during deletion. During repeated creation and deletion , this reference count became negative resulting in error. This has been fixed by incrementing default tpid reference count upon deletion of tpid thus providing support for repeated creation and deletion of tpid on a port
SDK-57105		56850_A2	The customer requested configuration of RTAG7_HASH_CONTROL_4.VXLAN_PAYLOAD_HASH_SELECT_A/B to meet their hash requirement. For Trident2 and subsequent XGS devices, 2 switch controls bcmSwitchHashVxlanPayloadSelect0 and bcmSwitchHashVxlanPayloadSelect1 have been provided to support the requirement.
SDK-57107		56850_A2	The customer requested configuration of RTAG7_HASH_CONTROL_4.VXLAN_PAYLOAD_HASH_SELECT_A/B to meet their hash requirement. For Trindent2 and subsequent XGS device, 2 switch controls bcmSwitchHashVxlanPayloadSelect0 and bcmSwitchHashVxlanPayloadSelect1 have been provided to support the requirement.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57123		56850_A0 56850_A1 56850_A2	Issue:- bcmSwitchL3Max128BV6Entries switch control setting caused assertion failed message due to array index overflow in array defip_tcam_log_index and defip_tcam_urpf_log_index of SOC_CONTROL. Fix:- Modified the soc_trident2_mem_config function to make sure l3_defip_index_remap won't exceed the physical size, and the arrays can be initialized after that.
SDK-57132	757170	88650_B1 88660_A0 88670_A0	OAM: Packets trapped by the OAM classifier with an incorrect level by an up-MEP will include two sets of system headers. The inner set will include the DSP, SSP on the FTMH, as well as a PPH and FHEI, the outer set will include a FHEI with the CPU-Trap-Code field set to 0xa2 (bcmRxTrapOamLevel). Similarly for packets that arrive at an up-MEP from the passive side. In this case the CPU-Trap-Code on the outer FHEI will be 0xac (bcmRxTrapOamPassive) and the inner set of system headers will be as above. This behavior may be enabled by setting the soc property "custom_feature_oam_additional_FTMH_on_error_packets" to 1.
SDK-57133	748626	88650_A0 88660_A0	When ilkn_tdm_dedicated_queuing feature is enabled, non-TDM ports can't reach wire speed. (blocked in ~60G). Fixed.
SDK-57141	779921	56840_A0 56850_A2	Problem: Ipbm mask setting was missing during field entry movement, which gets called when a higher priority field entry is installed. Solution: Ipbm mask was set properly during field entry movement for Trident Series of devices.
SDK-57164	763730	56440_A0 56450_B0	In the earlier release the tag information derivation was incorrect for PPD_TYPE=2, for PPD_TYPE=2 the tag information is present in packet itself and get derived through it, HG header contain tag information in case of PPD_TYPE=0 and 1 only where the outer tag get stripped out and added in the hg header, while the inner tag is still derived from the packet itself. This issue has been addressed and fixed in this release.
SDK-57184 SDK-57276		88660_A0	Bug found and fixed in BCM command diag prge_last causing "default null" program to be incorrectly printed.
SDK-57187	776877	56440_A0	For Katana2, bcm_cosq_gport_bandwidth_set was not setting the l2 shaper properly. As part of the fix l2 shaper will be configured properly.
SDK-57188	780510	56450_A0 56450_B0	bcm_mpls_port_add allocates two VPs in VPWS case, one for access and one for network port. But when bcm_mpls_port_add was invoked for second time to add network port to VPWS a new VP was being allocated instead of reusing the already allocated VP. Added fix to not allocate new VP if already allocated.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-57199		88650_B1	IMPORTANT: DEFAULT BEHAVIOR CHANGE FCoE packets were dropped when FCoE switch was not enabled (bcm886xx_fcoe_switch_mode = 0). From now on, FCoE packets are treated as Ethernet packets when FCoE is disabled.
SDK-57201	779706	88650_A0 88650_B0 88650_B1 88660_A0	STG: STG APIs create/destroy STGs and set/get spanning tree status of ports in STGs. Certain STG APIs (bcm_stg_create_id/ bcm_stg_destroy/bcm_stg_stp_set/ bcm_stg_stp_get/bcm_stg_detach) didn't release the mutex when existing with a non-zero value. The issue detailed above can cause deadlock when using certain STG APIs. Mutex can be correctly released after the fix.
SDK-57204		88650_A0 88650_B0 88660_A0	a fix to allow future ISSU capability for alloc manager
SDK-57207	777630	56640_A0 56640_A1 56640_B0	Issue:Packets of size 64 to 75 bytes getting dropped for XE ports. Root Cause: The runt threshold value for XE ports was getting set as 76 instead of the correct value 64. Hence packets of size 64-75 bytes were getting dropped. Fix: For Triumph3 and Katana2, put explicit checks to ensure that runt threshold value is set to correct value, i.e. RUNT_THRESHOLD_XE = 64, RUNT_THRESHOLD_GE = 64 and RUNT_THRESHOLD_HG = 76. Also optimized the function mac_x_init for multiple READ and WRITE for XMAC_RX_CTRL and XMAC_TX_CTRL. Added a single write common for all devices instead of multiple instances as was present previously.
SDK-57215		88650_A0 88660_A0 88670_A0	Trill multicast adjacency BCM API implemented with new APIs: bcm_trill_multicast_adjacency_add/delete bcm_trill_multicast_adjacency_delete_all bcm_trill_multicast_adjacency_traverse Example can be found in cint_trill.c file in function mult_adjacency.
SDK-57220	780270	56850_A0 56850_A1 56850_A2	When programming MPLS_ACTION_IF_BOS=0x5(0x5 = L3_ECMP) for a given MPLS label, the next hop entry type was set to be 1 for sending out the regular L3 packet in the previous release. In this release, the next hop entry type is set to be 0.
SDK-57224	780313	56850_A0 56850_A1 56850_A2	In earlier releases, BCM_L2_REPLACE_DES_HIT_CLEAR flag was not supported in XGS devices. This has been supported. This flag only can reset the HITDA field in L2_ENTRY table.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57230	758870	88660_A0	VLAN: L2 FECs can be used either for protection or to group LIFs like in the case of the PON application, in which the flag BCM_VLAN_PORT_FORWARD_GROUP is applied at bcm_vlan_port_create(). Removal of a L2 FEC using bcm_vlan_port_destroy() for a protection FEC, is performed at once for both the working and the protecting FECs upon removal of the Working path. This logic was applied also in cases where the FEC wasn't used for protection as in the case of FORWARD_GROUP, but only the specified FEC was removed as only one FEC is used for this type of applications. This logic caused FORWARD_GROUP FECs with odd id number not to be deleted as if they represent a protecting path. This was fixed, so that the protection working/protecting state condition upon FEC removal is applied only for protection FECs.
SDK-57235		56340_A0	On every DMA interval the counter value are read from FT_EXPORT_FIFO table and the value is populated in the report. On every read the value gets reset leading to the cumulative values not getting retained in the END report when the flow expires. With this release read of the registries clear of counter values is avoided to retain the cumulative value for the END report.
SDK-57239	778949	88650_B0	OAM: The following bugs have been fixed: When updating endpoints with bcm_oam_endpoint_create() with the BCM_OAM_ENDPOINT_REPLACE flag set, the SW DBs were incorrectly updated causing subsequent calls to bcm_oam_endpoint_destry() to fail. Similarly for BFD endpoints of type bcmBFDTunnelTypeUdp, multi-hop.
SDK-57245	781014	56450_A0 56450_B0	FLEX_CTR_BASE_COUNTER_IDX and FLEX_CTR_POOL_NUMBER were not being restored during mpls entry replace operation. Added fix to restore the FLEX counter fields and update during replace operation.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-57258	779428	`56450_A0 56450_B0`	Release notes from all sub-task JIRAs == A) SDK-57533 KT2 (Unicore and Warpcore) Several issues with configuration 12 (issue 1) == Problem 1) Null get was returning value 0 which is valid phy master value (i.e. slave). Fix 1) Added dummy `get_master()` in xgxs16g1l driver. Code is changed to return `MS_NONE` and that makes phy -master as NONE Problem 2: Proper medium was being detected in WARM-BOOT scenario only Fix 2: Corrected Copper/Fiber Medium detection concern in WC driver(which was applicable in Warmboot case only) Removed surrounding warm-boot condition in init part and now correct medium is returned Problem 3) Speed 1G was not advertised when port comes up as HG port and later converted to XE port due to max-speed set to >10000 Fix 3) Corrected WC 1g speed issue by checking additional `XE_PORT` type along with current speed before advertising speed Problem 4: TR 19 issue was happening with medium fiber i.e. test case was forcefully setting speed to 1G Fix 4) Added WC driver name check before forcing speed to 1G and by-passed concern == B) SDK-57534 KT2 (Unicore and Warpcore) Several issues with configuration 12 (issue 2) == Problem 1) Problem was happening due to wrong use of portgroup config variable in init phase and auto portgroup creation in flex-io operation i.e. assumed RXAUI related port group setting while converting hg port to 2 lane XE ports. Fix 1) Removed auto portgroup creation decision in flex-io operation and now user needs to set portgroup prior to flex-io operation and SDK initialization accordingly. If User doesn't use `auto_portgroup` config variable and doesn't set portgroup config variable prior to flex-io operation, SDK will throw "Behavior not guaranteed" message == C) SDK-57535 KT2 (Unicore and Warpcore) Several issues with configuration 12 (issue 3) ==

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57263	774859	88650_A0 88650_B0 88650_B1 88660_A0	In some cases when using the diagnostic 'diag pp pkttm', the meter pointer assigned to the packet would be displayed as invalid, even when the meter pointer assigned to the packet was valid. This is now fixed.
SDK-57270		88650_A0 88650_B0 88660_A0	Field Processor: Redirecting at egress according to a GPort of type System-Port was not supported. This is fixed. Reflector: The function `setup_port_for_reflector_program()` in `cint_benchmarking_methodology.c` has been changed so that the Egress FP rule modifies only the out-TM-port (by calling only the bcmFieldActionRedirect without bcmFIeldActionStat actions). For a more detailed account, see `cint_benchmarking_methodology.c`
SDK-57272		88650_A0 88650_B0 88660_A0	Diag pp dblif used to return 0 for the `has_cw` (in case lif is pwe) with no relation to the real value of. Now, it is returned depending on the real value.
SDK-57277	780887	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	Issue:- In parallel mode, if VRF=0, then hardware looks only in global bucket space for bucket match, so route with VRF=0 is not allowed to be inserted to ALPM table. But the examination code was not working for the first VRF=0 route insertion. Fix:- Adding VRF=0 is disallowed explicitly in parallel mode. Update the document for this restriction.
SDK-57283		88650_A0 88650_B0 88660_A0	There was a value mismatch between set and get by calling `bcm_switch_control_port_set/get` APIs, where type=bcmSwitchHashIP4Field0. This mismatch is fixed.
SDK-57289	779367	88650_B1 88660_A0	When using external TCAM, control-plane writes to the external TCAM could sometimes fail when performed during line speed traffic. This issue is fixed by setting "CpuRecordPrio" field in register "TransmitCfgs" to '1' in the external TCAM application initialization.
SDK-57290	781195	88650_A0 88660_A0	Fix `bcm_petra_trill_port_delete` functionality. Add calling of `_bcm_dpp_mc_to_trill_remove` function, that removes sw db `mc_id` to nickname.
SDK-57333	739837	56850_A0 56850_A1 56850_A2	Issue:- In previous implementation for BST index resolution, if cosq value -1 was used as input, cosq 0~7 were used to retrieve the index. but by default the max cosq number is 3. So the insertion was triggered. Fix:- replace cosq 0~7 by 0 ~ `COS_MAX(unit)` - 1.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57341	780620	56649_A0	When using "bcm_l2_learn_port_set" API to enable Class Based Learning for a trunk port, the function was returning error even though hardware programming was successful. This was because the API was trying to access another table which is not relevant for trunk ports and was using mod id value "-1" for this. The issue was resolved by adding an early return after programming the relevant Trunk table.
SDK-57343	782070	56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0	Communication between aging thread and other l2_addr_delete APIs thread is synchronized by binary semaphore. Ocasionally when the aging thread was stopped and restarted, there was a mismatch between semaphore give and take between aging thread and other API threads. This has been fixed.
SDK-57349	781836	88650_B1 88660_A0	L3 VRRP: In some cases, if there was an error in the l3 vrrp APIs, the L3 mutex was not released. The error has been fixed, and the mutex will always be released.
SDK-57354		56840_A0	After clear operation through bcm_esw_l2_clear(), the data in structure _bcm_l2_match_ctrl was sometimes released while the background thread L2MOD still needed to refer to the invalid data. This sometimes led to a crash of L2MOD. Currently the data in _bcm_l2_match_ctrl won't be released in bcm_esw_l2_clear() in order to avoid this race condition.
SDK-57434		56850_A0 56850_A1 56850_A2	In previous releases, memory write operation to Ingress Pipeline tables during a ING_RESET_CONTROL Operation causes inadvertent writes to L3_TUNNEL, UDF_CAM and ING_FC_HEADER_TYPE Tables. In this release, a new flag SOC_F_MEM_CLEAR_HW_ACC indicating whether ING_HW_RESET_CONTROL is used to clear a table was added. ING_HW_RESET_CONTROL action will only happen during system initialization. In any other cases, table clear is done via table SLAM operations.
SDK-57437		56850_A0	api bcm_vxlan_port_delete is working.
SDK-57459	782198	88650_B0 88650_B1 88660_A0	Fixing memory leak issue in TRILL. Destroy TRILL port didn't free allocated memory (Add BCM_FREE to _bcm_dpp_mc_to_trill_remove function).
SDK-57462	757100	88650_A0 88650_B0 88650_B1 88660_A0	Fixed l2 show diagnostic output for VPLS interface.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57469	780971	88650_A0 88650_B0 88650_B1 88660_A0	Add support for split-horizon for MPLS-Tunnel-initiator. This is useful when PWE label is built using EEI (label+push profile) and outlif that points to the EEDB is MPLS-Tunnel-initiator. In this case the PWE inherits it's orientation (HUB/ SPOKE) from the next tunnel. To set the orientation of MPLS tunnel use `bcm_port_class_set` with class=bcmPortClassForwardEgress and port=mpls tunnel gport.
SDK-57470		88650_A0 88650_B0 88650_B1 88660_A0	Reflector (RFC-2544): Etherner Reflector program (Swaping MAC adresses) has been updated to support double tagged packets. IP program will only support single tagged packets.
SDK-57476		56850_A0 56850_A1 56850_A2	In earlier releases `bcm_stat_group_create` could get stuck in loop for egress SVP counters under scaled set-up. The issue was due to macro `FLEX_COUNTER_DEFAULT_EGR_DVP_ATTRIBUTE_1_TABLE_POOL_NUMBER` not being defined correctly for TD2, which led to endless loop when the egress flex counter pool were exhausted. It was defined to 5 for all the chips include TD2 but actually it should be less than 4 for TD2 as TD2 only has 4 egress flex counter pools. The fix was to define separate macro for TD2.
SDK-57487		56850_A0 56850_A1 56850_A2	Previously, 1-bit error reporting enabling logic and SER correction logic for all MMU tables that are CPU accessible on TD2 was not fully implemented. They have been implemented in this release.
SDK-57498	783084	56450_A0 56450_B0	In `bcm_qos_map_create` API an untagged PHB variable was being used uninitialized and that resulted in an unexpected `ING_UNTAGGED_PHB` entry being created. This was fixed toprevent untagged PHB variable from creating an unexpected `ING_UNTAGGED_PHB` entry.
SDK-57500	783310	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In the previous release, the CPU port was not removed when the API `bcm_multicast_egress_delete_all` was called on Trident2. In this release, this issue has been addressed by removing the CPU port when the API `bcm_multicast_egress_delete_all` is called.
SDK-57503		56340_A0	Problem: `bcm_regex_policy_policer_attach` results in a crash because of internal compatibility check being done between level0 and level1 meters. Solution: Hierarchical meters are not supported on regex policies. Hence the compatibility check is disabled until we support hierarchical meters.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57505	783296	`88650_A0`	Fixed packet loss related to Reset CMIC interface in soft reset sequence.
SDK-57507	776846	`88030_A0 88030_B0`	Modification to `CORE_PORT_MODE` & `PHY_PORT_MODE` must be made with MAC in reset. Not following this rule may leave MAC in a state that no packets can be received or all received packets are runts. Other than this symptom, the MAC doesn't report anything wrong.
SDK-57515	780895	`88650_A0 88660_A0`	In L2 learning, traversing over the MACT to get all the inserted entries while learning, may result in an infinite loop in some rare cases. This is due to mis-handling of a rare state in an internal buffer. This is fixed.
SDK-57525	782992	`56850_A0 56850_A1 56850_A2`	Customer requested a mechanism to find out the entropy label used for a given vxlan flow. In order to provide the requested mechanism, `bcm_switch_pkt_info_hash_get` has been modified to return the entropy label used if the packet is for the vxlan. For packets encapsulated into VxLAN tunnels, Entropy label is generated using RTAG7 hash. By using `bcm_switch_pkt_info_hash_get(),` entropy label used for VxLan can be retrieved. Note that entropy label is piggy backed in `dst_intf.` Example) `hash_info.flags = BCM_SWITCH_PKT_INFO_HASH_UDP_SOURCE_PORT` print `bcm_switch_pkt_info_hash_get(unit, &hash_info, &dst_gport, &dst_intf);`
SDK-57533		`56450_A0 5645_A0 56450_B0 56456_A0 56455_A0`	Problem 1) Null get was returning value 0 which is valid phy master value (i.e. slave). Fix 1) Added dummy `get_master()` in xgxs16g1l driver. Code is changed to return `MS_NONE` and that makes phy -master as NONE Problem 2: Proper medium was being detected in WARM-BOOT scenario only Fix 2: Corrected Copper/Fiber Medium detection concern in WC driver(which was applicable in Warmboot case only) Removed surrounding warm-boot condition in init part and now correct medium is returned Problem 3) Speed 1G was not advertised when port comes up as HG port and later converted to XE port due to max-speed set to >10000 Fix 3) Corrected WC 1g speed issue by checking additional `XE_PORT` type along with current speed before advertising speed Problem 4: TR 19 issue was happening with medium fiber i.e. test case was forcefully setting speed to 1G Fix 4) Added WC driver name check before forcing speed to 1G and by-passed concern

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57534		56450_A0 5645_A0 56450_B0 56456_A0 56455_A0	Problem 1) Problem was happening due to wrong use of portgroup config variable in init phase and auto portgroup creation in flex-io operation i.e. assumed RXAUI related port group setting while converting hg port to 2 lane XE ports. Fix 1) Removed auto portgroup creation decision in flex-io operation and now user needs to set portgroup prior to flex-io operation and SDK initialization accordingly. If User doesn't use auto_portgroup config variable and doesn't set portgroup config variable prior to flex-io operation, SDK will throw "Behavior not guaranteed" message
SDK-57539	781348	56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0	Logical table Register settings of ACL_L2IP4_ONLY partition of external TCAM are not accommodating source mac and destination mac addresses in final key. In this JIRA, LTR settings of ACL_L2IP4_ONLY are modified to have source mac and destination mac at right offsets in final key.
SDK-57543	781991	56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56850_A0 56855_A0 56843_B0 56841_A3 56846_A1 56841_B0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In the previous release, when L2_MOD_FIFO mode was used and station movements happened, only one "ADD" notification would be issued on TD+, which was not incorrect. In this release, this issue has been improved by notifying one ""DEL" notification and one "ADD" notification in this kind of situation.
SDK-57548	783511	56850_A0 56850_A1 56850_A2	It was reported that all packets appear to be store and forward on the port when the INIT_VALUE was set to 0x3 during chip initialization for 1G mode. The issue was fixed by modifying the egress credit to 12 for all the speeds lower than 10Gbps.
SDK-57550	777385	56450_A0 56450_B0	THDO_QCONFIG_CELL could not be configured for packet processing ports greater than 128. This issue has been fixed to support complete range of packet processing ports(sub ports).
SDK-57556		56850_A0 56850_A1 56850_A2	corrected the test script by add the flag BCM_IPMC_RPF_FAIL_TOCPU to set value of IPMC_EXPECTED_L3_IIF_MISMATCH_TOCPU to 1

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-57558		56850_A0 56850_A1 56850_A2	In previous releases, bcm_vxlan_stat_detach still took high execution time because of some unnecessary memory operation and extra overhead. In this release, we cut some unnecessary memory operations and redundant codes to save time, therefore execution time are reduced.
SDK-57571		56540_A0 56540_B0	On TR3 device, if the number of COSQs is changed from default value (4) to 8, after the warm boot recovery, the number of COSQs still shows as 4, since this information is not stored in the persistent storage (scache). The issue is fixed by storing this information in scache and retrieving it during warm boot level 2 recovery.
SDK-57582	782398	56624_A0 56224_A0	When using autosync for warmboot, in case of remote link down event, the scache state was not being synced to the current state. So the link state might not be recovered properly during warmboot. This is fixed by marking the scache state as dirty when a remote linkdown event occurs, so that the state is synced later. Also, link_mask2 is not being recovered properly after warmboot. Now, this will be stored in scache and will be recovered during warmboot.
SDK-57584		88650_A0 88650_B0 88660_A0	BFD: When calling bfd_endpoint_create() with type== bcmBFDTunnelTypeMpls an additional TCAM entry is needed. Due to limited resources only 128 TCAM entries may be used for OAM/BFD. This JIRA verifies that this amount has not been exceeded and that TCAM indexes used are in the range 0-127.
SDK-57600	780870	88650_A0 88660_A0	Add push profile free when deleting PWE. Fixes resource push profile exhaustion when adding several MPLS tunnels and PWEs.
SDK-57625	786811	56340_A0	bcm_cosq_gport_attach is returning failure for some ports (RESOURCE_UNAVAIL) due to unavailability of L0 nodes.scheduler list is not reset completely on mmu soc reinitialize. Fixed the scheduler list reset.
SDK-57627	787010	56450_A0	Provided the API sequence on how to configure the burst rate
SDK-57630		88660_A0	OAM: fixed the loss_farend/nearend fields to return correct values (expressed in 100th of percent) in bcm_oam_loss_get()
SDK-57648		88650_A0	A new ISSU error detection mechanism is inserted to test if the variable layout in external storage has been changed between two versions without a proper ISSU handling. The code is under a new compilation flag BCM_ISSU_SANITY, in order to be used mainly internally for regression and debug and to be transparent to the user. If the compilation flag is used, both the old and the new versions must be compiled with that flag. NOTE that ISSU is not available if versions were compiled with different compilation flags.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57650		88650_A0 88660_A0	Failover ID values '-1' & '-2' are reserved for FEC Protection as 'No Protection' and 'Facility Protection' respectively. Those reserved values failed with an internal error. Now those values, when used with the BCM_FAILOVER_WITH_ID flag, produce an error that states that the value is out of range.
SDK-57652	781357	88650_A0 88660_A0	Ring Port: There was an error in bcm_vlan_port_find() where the returned failover_port_id was incorrect for a G.8032 Ring Port. The error has been fixed, and the failover_port_id is now returned correctly.
SDK-57663	783147	All 56846_A0 56844_A0 56846_A1	In earlier releases, field groups auto expansion was not recovered in Level 2 Warm Boot. In this release we now allocate an unused bit in scache layout of field module to store the groups auto expansion capability.
SDK-57669	770442	88650_A0 88650_B0 88650_B1 88660_A0	Added validity check that returns an error when user configure cos profile that is > 16 for PWE P2P.
SDK-57689		88650_A0 88660_A0	1. Reference only - added port tables to the reference list of SER-memories, which enables SER handling for these tables 2. Added dynamic memory indication for some memories. Refer to JIRA item SDK-56498 for details.
SDK-57691		88650_A0 88650_B0 88650_B1	bcm shell command "diag ssdump" was disabled. It's now enabled.
SDK-57707	787634	56640_A0	For some MACs in L2 cache, BPDU flag was not being set. This was causing ports to drop BPDUs when in STP blocked state. This issue was due to overwriting of flags for these MACs. Corrected the flags to CPU | BPDU.
SDK-57725	788276	88030_A0 88030_B0	src/appl/diag/ledproc.c: Change the previous common code into C3 dedicated code, in order to not affect other modules.
SDK-57739		56850_A0 56850_A1 56850_A2	add a new case AT_L3MC_Rep_009 to verify the flag of BCM_IPMC_RPF_FAIL_DROP
SDK-57743		56850_A0 56850_A1 56850_A2	In previous release, five variables were calculated based on stat_counter_id, and three of them were used as subscript to access arrays without checking their legal ranges. Therefore, memory access violation happened. This problem has been resolved through adding proper check to those parameters to ensure the validity of their values.
SDK-57744	787141	88750_A0 88650_A0	bcm_fabric_link_status_get retrieves several link status indications. some of these link indications are sticky and should be cleared. This indications changed to be cleared on read. Meaning that this API retrieves the status since the last call.
SDK-57745		88650_A0 88660_A0	In Field Processor, the validation of the action size was changed incorrectly, such that it verifies that the size is smaller the MAX size instead of smaller or equal to it. This is fixed.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57749		56340_A0 56640_A0	BCM56640, BCM56340 support Software Aging. The L2 entries are aged out if HITSA and HITDA are both 0. New capability is added to age out entries based on ONLY HITSA and not consider the HITDA. This is done by setting the config property '12x_age_only_on_hitsa' to 1.
SDK-57751		56340_A0 56640_A0	BCM56640, BCM56340 support Software Aging. The L2 entries are aged out if HITSA and HITDA are both 0. New capability is added to age out entries based on ONLY HITSA and not consider the HITDA. This is done by setting the config property '12x_age_only_on_hitsa' to 1.
SDK-57769	784039	56334_B0 56334_A0	In previous SDK releases, there are no SER correction support for several MMU blocks on Enduro, and thus once a parity error occurs in these blocks, it cannot be corrected and the error will be detected continuously. Fixed overview: The feature of SER correction for these MMU blocks on Enduro have been implemented. In addition, SER injection function has been added as well.
SDK-57774	781863	56850_A2	In the previous release, when the deleted l3 interface which had been added into the multicast group was re-created with the same id, the l3 interface was not attached to the multicast group, which was incorrect. In this release, this issue has been addressed by ensuring that the next hop index will be allocated and de-allocated by multicast module when the l3 interface's encapsulation id is added into and deleted from the multicast group.
SDK-57775	788841	56150_A0	Fixed possible race condition in SOC initialization routines.
SDK-57791	774941	88650_A0	In Policer module, in some cases bcm_policer_create fails incorrectly when the mode is bcmPolicerModeCoupledCascade, due to an internal software usage of an uninitialized structure. This is now fixed.
SDK-57802	788015	88650_A0 88650_B0 88660_A0	Fixed failure when deleting MPLS label in ILM table when using bcm_mpls_tunnel_switch_delete and SOC property 'mpls_termination_label_index_enable=1'
SDK-57812		88650_B0 88660_A0	When using external TCAM for ACL and/or forwarding databases, its configuration was not restored after warmboot. A preliminary support is added to restore external TCAM configuration during warmboot.
SDK-57813		88650_A0 88650_B0 88650_B1 88660_A0 88670_A0	In previous version, BCM_VLAN_PORT_WITH_ID wasn't working in forward group.now support this function.

Table 117:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
SDK-57821 SDK-58760		88650_A0 88660_A0 88670_A0	BFD: In bcm_bfd_endpoint_create() the field remote_flags was changed to monitor the Flags field on incoming BFD frames (as opposed to monitoring Flags in bcm_bfd_endpoint_get() as well as setting the Flags on outgoing packets). The field local_flags was added and is used to control the Flags on outgoing BFD frames and (this is consistent with fields such as remote_state/local_state, remote_daig/local_diag, etc.).
SDK-57828 SDK-56669		88650_A0 88660_A0	BFD: addition of the filed loc_clear_threshold for bcm_bfd_endpoint_create(). This determines the amount of BFD frames received by the OAMP before a loss of continuity is cleared and a bcmBFDEventEndpointTimein event is triggered. This may be set at 0,1,2,3. Default behavior remains unaffected.
SDK-57844		56850_A0	In earlier release, adding one more IPV6_64B entry to table already with full IPV6_64B entries and some free IPV6_128B entries would result in inconsistency in software tables. Then trying to insert another route with same prefix would cause the process to fall into an infinite loop. This issue has been resolved.
SDK-57853		88660_A0	Trill warmboot. Sw state trill alloc link list size was not correctly calculated at warmboot trill restore, causing incorrect size after warmboot
SDK-57863		88650_A0 88660_A0	BFD: Adding accelerated endpoint with bcm_bfd_endpoint_create() while in local_discr field any of the bits 13-15 is set, caused error.
SDK-57866		88650_A0	In Rx trap module, in the allocation of a programmable trap, the error validation was incorrect. Fixed.
SDK-57889	789898	56340_A0	Aging thread runs independently , wait on semaphore for age time and wakes up & runs the bulk operation. L2 bulk operation stops and starts the aging thread. Here aging thread does not account for elapsed time it has already spent on semaphore.Lets say, aging thread already spent 25 sec and port delete operation is called at that time, aging thread stopped and restarted. Here MAC will be deleted at (25 + 30) sec = 45 sec. This is adjusted by keeping a log of elapsed time on semaphore during aging thread exit and readjusting it, during next time aging thread started.
SDK-57911	789281	88650_A0	The previous is that pass the NULL pointer of uc/mc/bc for _bcm_petra_vlan_flooding_per_lif_get() caused the segment fault. The fix is that pass the uc/mc/bc for _bcm_petra_vlan_flooding_per_lif_get() and get the value by the uc/mc/bc.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57942		88750_A0 88750_B0	When using BCM88750 repeater, due to miss-configuration some corrupted cells might be dropped at the repeater ingress while it should be dropped at the destination device. Fixed.
SDK-57952		56640_A0	In TR3 device we were not able to delete last set of vlan service queues in the given port. We are now able to delete all the vlan service queues.
SDK-57958	782029	88650_A0 88650_B0 88650_B1 88660_A0	MPLS bug fixed: when creating and deleting a tunnel label and PWE WITH_ID, if they are re-created and the lif ids swapped, the tunnel termination fails and the tunnel label is used for forwarding. Issue is due to uninitialized values (Destination and Destination valid) when setting MPLS tunnel termination in LIF table.
SDK-57961	777725	56640_B0 56850_A2	V4 routes take half entry in shared defip tables. There can be some defip indexes containing V4 routes that does not have other half filled due to prefix restrictions. When SDK tries to recover number of indexes used for V4 prefixes in defip tables, it divides total number of routes by half but does not account about half entries. Due to this, SDK gives wrong number for total available 64/128 V6 route entries. This lead to overwriting of already existing routes in the defip tables. The fix is to count the number of half entries in defip tables and then use them during derivation of total free entries left in defip table for 64V6 and 128V6. This way SDK can return BCM_E_FULL error at table full and will not overwrite the existing entries
SDK-57962	747614	88650_A0 88650_B0 88650_B1 88660_A0	In L2 MAC table, when using API bcm_l2_addr_delete_by_vlan_port() to flush all the entries on a specific PWE, all the entries that are on the same MPLS-FEC as the specified PWE were flushed. This is fixed and now only entries that are on the specific PWE (PWE-label+MPLS-FEC) are flushed.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-57969	790396	56850_A0 56850_A1 56850_A2	In earlier releases, bcm_ipmc_traverse did not get IPMC v6 entries after level 2 warmboot. "ipmc ip6table show" shows nothing, but the HW table L3_ENTRY_IPV6_MULTICAST showed all installed entries. There were two root causes as follows: 1. The first L3_IPMC entry (index == 0) was reserved for default behavior for all chips except Katana and Katana2. In cold boot, we can find it was initialized in routine bcm_esw_ipmc_init. But routine _bcm_tr_ipmc_reinit considered it as normal L3_IPMC entry. 2. Flag of BCM_L3_IP6 was not set when recovering L3_ENTRY_IPV6_MULTICAST to l3 entry in warmboot. This caused the 'ipmc ip6table show' not work since low level driver considered it as IPV4 entries. The solutions are as follows: Routine _bcm_tr_ipmc_reinit has been changed to reserve first L3_IPMC entry for the default behavior for both Kanata and Katana2. Flag BCM_L3_IP6 has been set in recovering L3_ENTRY_IPV6_MULTICAST to l3 entry in warmboot in this release.
SDK-57975	790586	56450_B0	VLAN parameter check was being verified for both VPLS and MIM VPNs, though the VPN is of VPLS type. Fixed to validate VLAN only for the matching VPN type VPLS or MIM accordingly.
SDK-57977	769739	88650_A0 88650_B0 88650_B1 88660_A0	In Ingress Field Processor, cascaded Field groups are using the bcmFieldActionCascadedKeyValueSet action to transmit an action value as part of the key value. This action is always performed in the HW. If unset, a zero value is expected to be transmitted. The entry TCAM action encoding was incorrect if: 1. The Field group ASET was including bcmFieldActionCascadedKeyValueSet 2. No action value was explicitly set for bcmFieldActionCascadedKeyValueSet (even a zero value) This is fixed.
SDK-57984		56640_A0 56640_A1 56640_B0	In earlier releases, after the successful warm boot recovery of VFP configuration with flex stat on TR3, if one of the field qualifiers is deleted, reinstalling of field entry failed with invalid parameter. The issue is now fixed by passing the right parameter (offset_mode) in the internal function while updating flex stat for the field entry.
SDK-57994		88650_A0	In L3 IP applications, the max VRF value is limited to 4095 in HW, but a segmentation fault was occurring when using a VRF larger than 255. This was due to a warmboot engine variable saving a VRF bitmap with an inappropriate size. Fixed. ISSU is handled transparently.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57998		56640_A0 56340_A0	For BCM56640 and BCM56340, if config property l2x_age_only_on_hitsa is set, aging occurs only on HITSA, ignoring HITDA. In accordance with this the behavior of BCM_L2_HIT flag (bcm_l2_addr_t structure) is also changed, when this config variable is set. If the config variable is not set, BCM_L2_HIT flag is set for the entries passed to callbacks which have either HITSA or HITDA set. Also, in cases when an entry is being added, if this flag is set in the input bcm_l2_addr_t structure, both HITSA and HITDA will be set for the added entry. If the config variable is set, BCM_L2_HIT flag is set for those entries which have HITSA set. In case adding an entry, if this flag is set in the input bcm_l2_addr_t structure, only HITSA is set. To set HITDA, BCM_L2_DES_HIT flag has to set explicitly. Similar changes are reflected in l2 show diag shell command.
SDK-57999		88650_A0	In L2CP traps set via bcm_l2_cache_set, the handling of L2CP trap index of type Multicast and ports was incorrect. This is fixed.
SDK-58006		88650_A0 88650_B0 88650_B1 88660_A0	Cint: cint_ip_tunnel.c. ip tunnel was created in cint with incorrect ttl and dscp. Caused by SDK-55162. Consequently, checking ttl or dscp values in ip tunnels were failing when using cint_ip_tunnel.c.
SDK-58016		56850_A2	In the previous release, when configuring same Virtual Port (NIV Port) under multiple Mirroring sessions to get multiple copies, the function "_bcm_xgs3_mtp_slot_port_indexes_get" was called with the input port" parameter set to be physical gport. In this release, this function is called with the input port" parameter set to be physical unit port in this situation.
SDK-58023	736724	88650_A0 88650_B0 88650_B1 88660_A0	Added support for Split-Horizon filter in AC P2P VSI service. Up until now AC P2P service set always as Spoke, now user can configure and decide to set it either as Spoke or Hub using BCM_VLAN_PORT_NETWORK flag.
SDK-58043	790661	56450_A0 56450_B0	mpls_port add for VPLS configuration was using uninitialized pp port for given modid and portid. Fixed to convert given modid and portid to appropriate pp port before actual use. Another issue was that during subport trunk add for mpls VPLS port source_trunk_map table for the required entry is not configured appropriately due to incorrect modid and portid use internally. Fixed to convert pp port to appropriate modid and portid to update the required entry in source_trunk_map table.
SDK-58046		56850_A2	Customer found that a flex counter was automatically attached after calling bcm_vxlan_port_add() API. The root cause was the information of the L3 view was not cleared after changing the entry type. Now the issue has been fixed.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-58047		56850_A0	Problem Statement: DMVOQ functionality is not working Incorporated all the required change in SDK to make DMVOQ to work, for ex: Programming E2ECC_TX_PORTS_NUM, Programming of INTFI_CFG, Programming of CONGESTION_STATE_BYTES , implementation to support higig trunk ,Warmboot changes, Hw_index allocation is not proper, handled all the error cases ,Corrected the programming of FC_MAP_TBL, FC_ST_TBL, MMU_INTFI_OFFSET_MAP_TBL
SDK-58081	791991	56850_A0 56850_A1 56850_A2	Customer found rate limit on broadcast traffic was affected by changing DLF rate limit on the same port on TD2. It was the expected behavior in the system but some special improvement could be implemented for TD2. Now an improvement has been added to reduce this impact.
SDK-58087	792172	56450_A0 56450_B0	Subscriber delete was failing as the corresponding entry in REPL_LIST list was not removed when there are no EGR_L3_INTF/ EGR_L3_NEXT_HOP interfaces valid for replication of subscriber traffic and instead a new entry with NULL interfaces was added. The user is now able to delete the subscriber replication entries after correcting the behavior to remove the entry from the replication list when there are no replication interfaces.
SDK-58102		88650_A0 88660_A0	OAM: The following bug was fixed: Protection packets might not be sent until an event is registered.
SDK-58119	792999	56850_A2	Issue :- Qualifying SrcVxlanGport in Lookup Stage returns Internal Error Fix :- During Stage Lookup Qualifiers init routine, assigning secondary Qualifier for SrcVxlanGport was missed out. Hence the issue. Added Secondary qualifiers for SrcVxlanGport
SDK-58132		88650_A0 88660_A0	In some cases in bcm88650_A0, bcm88650_B0 and bcm88660_A0 (when the SOC property RATE_COLOR_BLIND is set to 0) the driver would crash when calling bcm_policer_destroy_all. This is due to a software bug where uninitialized memory is used. This issue is now fixed.
SDK-58135		88650_A0 88650_B0 88660_A0	PWE bug fixed: When deleting mpls port, push profiles of the PWE is deleted as well as all the push profiles of the MPLS tunnels that the PWE is pointing on.
SDK-58136	760903	88650_B1	The original issue was that ARAD PCIe controller dropped the completion when accessing the null space CMIC descriptor address that caused the CMIC logic to wait for completion forever. The current fix is that we enable the ENABLE PURGE IF USERIF TIMESOUT in which case CMIC HW will inject a fake completion after timer expires and injects a ECRC error so that DMA engines will come out gracefully.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-58138	791882	88650_A0 88650_B0 88650_B1 88660_A0	PWE P2P: Model was always Uniform, no matter what user configuration is. This was fixed - both Uniform and Pipe models can be configured.
SDK-58140	792399	88650_B1 88660_A0	Prevented false alarm memory ECC errors from happening when using the direct base queue to modport mapping mode, and not configuring queue mappings for queues that receive traffic or credits for a scheduler.
SDK-58144	792827	56224_B0	Multicast init fails during warmboot in `bcm_esw_mcast_init()` on some flavors of RAVEN chipset which do not have native L3 support. Provided a runtime check in `bcm_esw_mcast_init()` to check whether L3 features are supported on chipset.
SDK-58173		56440_B0	Problem - IPMC replication of L3 Multicast traffic over L3 interfaces is not working. Two issues are observed 1. While adding the IPMC configuration `L3_IPMC MMU_MC_REDIRECTION_PTR` is being modified which is not intended action in `"bcm_esw_ipmc_add"` Removed the code to reconfigure the `MMU_MC_REDIRECTION_PTR "tr_ipmc_write"` 2. when subscriber replication on `EGR_L3_INTF` , `EXT_MC_QUEUE_LIST0/ EXT_MC_QUEUE_LIST1` field `L3_REPL_PTR0/1` is not programmed properly taking into account that `EGR_L3_INTF` index shall offset into (8k+ index) while configuring `L3_REPL_PTR.` It should have been programmed to be 8K+ `EGR_L3_INTF(index).` IPMC replication of L3 Multicast traffic over L3 interfaces is working after programming the index correctly in `EXT_MC_QUEUE_LIST0/ EXT_MC_QUEUE_LIST1` - field `L3_REPL_PTRx` for L3 interfaces.
SDK-58188	791948	88650_A0 88650_B0 88650_B1 88660_A0	In L2 MAC table, when using API `bcm_l2_replace_match()` with flag `BCM_L2_REPLACE_PROTECTION_RING,` the user supplied port mask was not used. This is fixed and now port mask is taken into consideration correctly.
SDK-58194		88650_A0	Fix bug in print encapsulation name in case of SOP only encapsulation. No effect on customer application.
SDK-58198		88650_A0	Fixed - `bcm_port_loopback_set()` with `BCM_PORT_LOOPBACK_PHY_REMOTE` for Arad fabric links. Fixed.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-58208		88650_A0	In external TCAM ACL, the actions are different between BCM88650 and BCM88660 due to the common IP database in BCM88660: - The first action (bcmFieldActionExternalValue0Set) can be used for forwarding (or ACL) in both devices, with size of 48 bits - The second action (bcmFieldActionExternalValue1Set) can be used for RPF (or ACL) in BCM88650, with size of 16 bits. In BCM88660, it can be used as ACL with size 32 bits. - The second action (bcmFieldActionExternalValue2Set) can be used for RPF (or ACL) in BCM88660, with size of 16 bits. In BCM88650, it can be used as ACL with size 32 bits. - The third action (bcmFieldActionExternalValue3Set) can be used as ACL with size 24 bits. The distinction between BCM88650 and BCM88660 devices was performed only according to the compilation flag (BCM_88660_A0) and not according to the unit type.
SDK-58237		56850_A2	Added 10G XFI FEC support.
SDK-58243	790578	88650_A0 88640_A0 88670_A0	FP: Fixed cases in which upper bits of the result were not initialized to Zeros while getting a Field.
SDK-58275		88650_A0 88660_A0	OAM: Following bug was fixed: When soc property "custom_feature_egress_snooping_advanced" is on and calling bcm_oam_endpoint_action_set() with the destination set as a trap, function may fail to properly update the classifier for up MEPs.
SDK-58315		88650_A0 88660_A0	Ring Protection: Added support for an optional sequence to perform Fast Flush. The sequence is comprised of encoding the Ring Port (FEC) as a gport of type FORWARD Port and calling bcm_l2_replace with BCM_L2_REPLACE_MATCH_DEST instead of BCM_L2_REPLACE_PROTECTION_RING.
SDK-58324		All	Fixed bug in KNET SKB allocation when using BCM API Rx buffers (use_rx_skb=0) on virtual KNET network interfaces which do not strip the VLAN tag.
SDK-58345		56850_A0 56855_A0 56850_A2	In the previous release, customer found the flex counters were not cleared after detaching. It was because on TD2, SDK only cleared X-pipe counters after detaching, but not Y-pipe counters. Now this issue has been fixed on TD2.
SDK-58358		88650_A0 88660_A0 88670_A0	Jericho Protection: BCM API added for all Jericho Protection enhancements. Separate Protection tables are implemented. For Arad, the usage of bcm_failover_create() is modified and it's required to specify a failover type by setting one of the failover type flags: BCM_FAILOVER_FEC, BCM_FAILOVER_INGRESS & BCM_FAILOVER_L2_LOOKUP. The failover ID itself is now encoded with a failover type.

Table 117:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
SDK-58360		`88750_A0 88650_A0`	`bcm_fabric_link_status_get` retrieve the fabric link status. This specific link status should be cleared on read. However, the entire quad status was cleared . Fixed.
SDK-58364	794704	`56850_A0 56850_A1 56850_A2`	In previous release, when customers changed one port encapsulation on TD2, the Rx Max Size was reset but Egress MTU is not. This was an omission and has been resolved.
SDK-58375	795815	`88750_A0 88750_B0`	Fixed a bug in "diag queues" command, the command previously printed a wrong occupied link number.
SDK-58382		`56243_B0 56243_A0 56242_B0`	When one port is moved to PHY loopback on `56243_A0,` it turns all 4 ports in loopback. Provided a protective check in `56243_A0` internal phy driver to use broadcast mode of PHY config only when lane 0 is disabled, else use single lane config mode.
SDK-58392		`88660_A0`	`bcm_port_enable_set` bug fix: In case that 2 CAUI ports and ELK are configured, The CGE1 traffic was dropped when the ELK port was disabled.
SDK-58393	794812	`56541_A0 56540_A0 56540_B0 56541_B0`	Wrong meter table size configuration in BCM5654X devices resulting in failure of `bcm_policer_destroy,` is fixed.
SDK-58395		All	1588/PTP servo configuration takes `bridge_time` parameter in struct `bcm_ptp_servo_config_s` which normally configured in seconds. Comments are modified to reflect the same.
SDK-58398	793706	`88750_A0`	SER interrupts were not signaled to CPU and not counted, due to being masked by a set of override bits called monitor bits. This was fixed to allow proper logging and handling of SER interrupt events by the SDK.
SDK-58407	769153	`56640_A0`	On TR3 device ,for HSP ports the sdk was not programming `HES_PORT_CONFIG` register , hence the scheduling was not proper .As part of the fix we are programming it properly .
SDK-58409		`56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0`	Previously, the software control structure for IBOD was not stored in scache, and hence during warmboot, the state was lost, which caused the workaround (IBOD WAR) to run again on all the effected ports. Now, the related fields in the control structure are stored in scache and are recovered during warmboot.
SDK-58410		`88650_A0 88660_A0`	OAM: when calling `bcm_oam_action_set(),` a newly allocated trap code will be returned via the field `rx_trap,` when applicable.
SDK-58423	793902	All `56850_A0 56850_A1 56850_A2`	In the previous release, the function `rx_higig2_vpn_resolve` didn`t parse VNI field in HIGIG2 header when it was VxLan. In this release, support has been added to parse VNI field in HIGIG2 header when it is VxLan.
SDK-58425	796179	`56440_B0 56450_B0`	For BCM5645x devices, support has been added for L3 packets, ingressing from trunk member ports, to be able to be trapped to CPU.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-58428	796509	`56450_B0`	Fixed the issue of CCM RX not working after deletion and re-creation of RMEP due to wrongful deletion of OAM lookup key as part of RMEP delete.
SDK-58445		`56640_A0` `56640_A1` `56640_B0`	Problem: When the cascaded TCAMs are present, and the lookups are configured to include L2, L3, and L2+ACL, the system will forward a few thousand packets and then hang when the ESM bandwidth is oversubscribed. This will manifest itself as an ETU response FIFO underrun. Solution: Number of lanes to the TCAM interface is different when there are multiple TCAMs present. Fixed the issue by setting the number of lanes based on the number of TCAMs present in the system.
SDK-58460		`88660_A0`	Routing Over VXLAN feature in `BCM_88660`. At ingress node, UDP length calculation was incorrect when sending a packet from TOR to Overlay network. API require to configure per native router interface, the expected amount of native Ethernet VLAN tags to be built at the native Ethernet header. BCM calling sequence: when creating the native router interface: `(bcm_l3_intf_create)`, fill the member: `native_routing_vlan_tags` from `bcm_l3_intf_t` with the expected amount of native Ethernet vlan tags to be built at the native Ethernet header. see `cint_vxlan_roo.c` for more information. See `cint_vxlan_roo.`

SDK-58461	794274	`88650_A0`
		`88650ACP_A0`
		`88650_B0` `88650_B1`
		`88660_A0`
			Problematic part is removed. there should be no limitation to 16 indices.

SDK-58462		`88650_A0` `88650_B0` `88660_A0`	Uninitialized variables were found in both PMF compare operation function and in LIF & RIF profile management function. This is fixed.
SDK-58463		`88650_A0` `88650_B0` `88660_A0`	In TCAM management, a sorted linked-list is used to handle the entry priority ranges: given a priority, the sorted list indicates the acceptable line range to insert a new TCAM entry. When looking to insert / get a node of this linked-list, the lookup function scans the list until it founds a node with lower priority and a node with higher priority. This lookup was not always correct. This is fixed.
SDK-58466		`88650_B0` `88650_B1` `88660_A0`	Required changes in SDK in order to have full support of external TCAM configuration restoration after warmboot. Note that this support requires the use KBP-SDK 1.2.5 and higher.
SDK-58472	793850	`88650_B1`	Remove ECC 1bit and 2 bit monitoring from ILKN memories of lanes that are not in use.
SDK-58487		`88650_B0` `88650_B1` `88660_A0` `88670_A0`	In Ingress Field Processor, for field groups of bcmFieldGroupModeDirectExtraction type, if an entry has a 1x1 mapping (action=extracted-field) then use action macro FES instead of FEM.

Table 117:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
SDK-58493		`88650_A0`	- Memory error protection: Enabling the ECC & Parity error mechanism sequence moved to the end of `soc_init` sequence. The reason is that this mechanism should be activated only when all memories are initiated, to avoid false alarms. - Shadowing (caching) mechanism: Memory shadows were not updated according to HW memory values after initialization sequence. Consequently, the cache was invalid until the appropriate entry was set explicitly following a driver call that translates to the corresponding table entry. Trying to use cache prior to writing the entry, e.g. as part of parity error-correction interrupt handler, would fail. The fix ensures that all the cache memories are updated with HW by default, as part of driver initialization.
SDK-58528	688507	`88650_A0 88650_B0 88650_B1 88660_A0`	OAM: Bug in OAM classification exposed only when adding the build option `DEBUG_OPTIMIZE=TRUE.` This bug causes incorrect oam module initialization when running in optimized mode.
SDK-58553	796964	`56744_A0`	In previous release, HG port with 25G speed config couldn't be added to config file. HG port with 25G speed config was enabled by modifying SDK to accept such configuration.
SDK-58565	797854	`56850_A2`	In the previous release, in `bcm_tr2_vlan_gport_add,` the `ing_port_bitmap` was overwritten by IPMC groups member port bitmap when we updated the `ing_port_bitmap` in `VLAN_TAB.` In this release, the `ing_port_bitmap` is read out first and then ORed with IPMC groups member port bitmap.
SDK-58568		`88650_A0 88650_B0 88660_A0`	In IPv6 Multicast FLP program, a new SOC property `(custom_feature_ipv6_mc_forwarding_disable)` is implemented. When this SOC property is enabled, the IPv6 Multicast FLP program is initialized with default values, where no key (and no lookup) is defined.
SDK-58596		`56850_A2`	Redirect to NIV virtual port from Ingress Stage Field Processor support was missing in earlier releases. The support is now added.
SDK-58630	792060	`56850_A0 56850_A1 56850_A2`	In previous releases, command "l3 ip6route show" would result in a cause when ALPM was enabled. This has been resolved by adding proper check to return feature unavailable instead of crashing when ALPM is enabled.
SDK-58649	792420	`88660_A0`	On Arad+, when calling `bcm_trunk_set` where all members have `BCM_TRUNK_MEMBER_INGRESS_DISABLE` set, an assert occurs. This assertion was fixed.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-58651		56850_A0 56850_A1	Issue :- Hashing is not happening when FP RedirectEgressNextHop action is used to forward packets to ECMP group. Solution:- When the IFP wants to route an IP packet to an ECMP group, it must use the L3_SWITCH action. This will ensure that the correct ECMP hash selectors are applied to the packet. (bcmFieldActionL3Switch) Only when the IFP wants to "route" a non-IP packet to an ECMP group, it must use the "Redirect to ECMP Group" action.
SDK-58667		56640_B0	a new enum value is added to enable the rx_lox external pin for 100G application
SDK-58695	799873	56450_A0	For BCM5645x devices, support has been added for L3 packets, ingressing from trunk member ports, to be able to be trapped to CPU.
SDK-58733		All	New build toolchain XLP SDK 3.0.2 effected on GTR and WRX based SVKs.
SDK-58752	783027	All	OAM: Support handling of CCM packets in the OAMP for packet types with 2 Ethernet headers (i.e. Mac in Mac). This requires correctly setting PPH.FWD_HEADER_OFFSET to the start of the OAM header.
SDK-58769		56450_A0 56450_B0	Problem statement: DMVOQ is not working when fabric is in between and trunk ports are used in fabric Fix description: Incorporated the changes so that DMVOQ works properly, when fabric is connected inbetween the modules and trunk ports are used to connect the egress module
SDK-58784		88650_A0 88650_B0 88650_B1	Extend Warmboot engine capabilities to support easily the removal of variables in future ISSU scenarios. This is not affecting ISSU in older versions.
SDK-58787		88750_B0	BCM88750 support running memory bist test on device initialization (enabled by soc property "bist_enable"). A block reset was added after memory bist test in order to make sure that memory is re-initialized.
SDK-58791		88650_A0 88660_A0	BFD: For Arad B1 and bellow allow all types of BFD Flags on frames transmitted from the OAMP. When calling bcm_bfd_endpoint_create(), the local_flags field may be any value between 0 and 0x3f.
SDK-58821	672112	88750_A0 88750_B0	"phy measure sfi" diagnostic command is used to measure fabric link rate. Using this command caused to the fabric link counters to stop. Fixed.
SDK-58863	797630	88650_A0 88660_A0 88670_A0	arp extension feature: Backward compatibility for arp extension feature. BCM_88660_A0 can now work in BCM_88650 mode using custom soc property custom_feature_next_hop_mac_ext_arad_compatible. Using BCM_88650 mode implies learning extension header always appended when PP packet is sent to the fabric.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-58870	800874	56850_A0 56850_A1 56850_A2	Provided the debug (FP+Verbose) prints to display the UDF chunks allocated during data qualifier create bcm_field_data_qualifier_create() and bcm_field_data_qualifier_get() API. It helps to understand the chunks used in the UDF offset during creation.
SDK-58893	798171	88030_B0	add per channel counters for "show counter" CLI command on bcm88030
SDK-58900		88650_B1 88660_A0 88670_A0	In Ingress Field Processor, in Direct Extraction field groups diagnostics, the source key selection for the action was read from wrong location. This is fixed.
SDK-58943		88650_A0 88660_A0	Allocation of MPLS push profile is now supported through bcm_mpls_port_add api. To allocate push profile, api should be called with BCM_MPLS_PORT_WITH_ID and BCM_MPLS_PORT2_TUNNEL_PUSH_INFO flags set. mpls_port_id is used to indicate the push profile. Function system_aux_push_profile_to_push_profile_id in cint_system_vswitch_encoding.c should be used to set the encoding of the id to be of type push profile. egress_label field and BCM_MPLS_PORT_CONTROL_WORD flag indicate the push profile properties. Example of usage can be found in vswitch_vpls_allocate_push_profile function in cint_vswitch_vpls.c.
SDK-58944		88650_B0 88650_B1 88660_A0 88670_A0	Documentation for cint_field_presel_advanced_mode added to readme file
SDK-58984	802028	56248L_A0 56248L_B0 56450_A0 56450_B0	The TTL value of VC label was not retained when replacing l3 egress object using API bcm_l3_egress_create() with "flags = BCM_L3_WITH_ID|BCM_L3_REPLACE". The issue has been fixed for BCM5645x and BCM56248L devices.
SDK-59142	804236	56063 56064 56062 56060 53400_A0 53406_A0 53404_A0 53403_A0 53402_A0 53401_A0	The RCPU support is aligned with the declared feature set in Make.local for Greyhound and Ranger2 devices in this release. In previous release, a warning message of "feature not available" was shown if RCPU was removed from feature set.
SDK-59202	763574	56854_A2	In the 6.3.7 release, SDK initialization would occasionally fail on BCM56854 devices. This has been resolved.
SDK-59532		88660_A0	OAM: In Arad+, when defining a MEP on a LIF on which another passive MEP or MIP resides, the second MEP may change the first MEP's non accelerated profile, thus changing the behavior of the first MEP.
SDK-59540		88660_A0	OAM: When using OAM classification in Arad mode on Arad+, an incorrect number of non-accelerated MEP-profiles may be allocated which may cause inconsistent behavior.

Table 117:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-59785		88650_B0 88660_A0	Required changes in SDK in order to support KBP-SDK 1.3.0 for external TCAM are introduced.
SDK-59954		88650_A0 88660_A0 88650_B0	De-allocation of scheduling elements fixes: 1. Non-composite HR SE couldn't be deleted. 2. FQ SE might deallocate from the wrong range, when range odd_even=1.

Section 13: Resolved Issues for 6.4.0

The following issues are resolved in version 6.4.0 of the SDK.

Table 118:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-46565	633504	`56334_B0`	In PTP/1588 application for Keystone processor there was an issue where setting VLAN priority <>"0" resulted in loss of communication between ToP and Host. In this release the VLAN priority mask has been corrected for Keystone.
SDK-47155	620527	`56440_A0`	In previous release, is, the `egress_tunnel_if` was only returned if the flag `BCM_MPLS_PORT_NETWORK` was set - which was incorrect because it should be set as well if `BCM_MPLS_PORT_EGRESS_TUNNEL` is used. In this release the following has been updated: retrieving `mpls_port->egress_tunnel_if` no longer depends on `network_port_flag.` We now check the `egr_l3_next_hop` entry type, if it is MPLS type, then we now set the `BCM_MPLS_PORT_EGRESS_TUNNEL` flag and retrieve the `egress_tunnel_if.`
SDK-47170	641741	`56440_A0`	During warmboot the SDK does not distinguish if the replication is on nexthop or L3 interface. This causes a warmboot failure when the replication is on nexthop. Fixed the warmboot logic to identify if the replication is on nexthop or L3 interface as per the configuration in HW replication table.
SDK-47774		`88650_A0`	In IP routing, the L3VPN-Default-Routing feature was not implemented: `BCM_L3_INGRESS_GLOBAL_ROUTE` had no effect upon calling. This is fixed: if the L3VPN-Default-Routing attribute is set, the IP routing lookups of the packet are <VRF, DIP> key and <0, DIP> if not found. No RPF check is performed.
SDK-47997	660499	`88030_A0`	The individual tests can now be configured to retain their configuration parameters upon termination of that test. That parameters will then apply to all subsequent tests. The default behavior is to roll back all the configurations. However, if it is desired to make a particular parameter persistent, it should have the line <cleanup> 0 </cleanup> on it.
SDK-49699	677743	`88030_A0`	New feature to support multiple cos levels and strict priority queue selection
SDK-49700	685812	`88030_A0`	The API `soc_sbx_caladan3_cop_policer_token_number_get()` is used to read token number of a policer.
SDK-49806		`88650_A0 88650_B0`	In PON application, in IPv6 Source bind implementation, the code has been changed to be more generic.

Table 118:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.0***
SDK-49819		`88650_A0`	Calling `soc_dpp_wb_engine_deinit` on one unit zeroed internal structs that contain information for all units in the system. problem is now FIXED, deinit will zero only structs belong to the specified unit.
SDK-49861		`88650_A0 88650_B0 88650_B1 88660_A0`	When working in MESH mode, VoQ must be mapped to a legal VoQ connector. Therefore addingdeleting a VOQ when it is under traffic is forbidden (these operations map the VOQ to an invalid VOQ). A verification was added in MESH mode, such that when voq is unmapped (a.e. mapped to an invalid connector), an error will be thrown if traffic still arrives to the VoQ. Note that this fix doesn't provide full protection, and is intended to catch an invalid state where possible. It is the application responsibility to make sure that the VoQ currently being unmapped doesn't receive any traffic.
SDK-49932	689754	`88650_A0`	In L3, in BCM886XX, the IPv6 host table is shared in TCAM with regular IPv6 forwarding table. However, `bcm_l3_host_add` API was supported for IPv6 but not `bcm_l3_host_remove` and `bcm_l3_host_find.` This is fixed.
SDK-50064	687256	`56643_B0`	MCSPRI was programmed with offset of 1024. The bit length of register MCSPRI is sufficient for absolute index and no offset is required. Fixed in by writing the actual index in the registry with no offset.
SDK-50087	690469	`88030_A0`	When immediate values are used for hstore they are checked for: • range (38 -256) • That the index plus the length does not exceed 256
SDK-50144	692372	`88750_A0 88650_A0 88750_B0 88650_B0 88650_B1`	In eyescan.h SOC function `soc_port_phy_eyescan_res_print` is no longer available for use. The print function has been moved to diagnostics shell, and is called from "phy diag eyescan" command.
SDK-50231	691831	`88030_A0`	A bug in the prior releases of the MDE manifested itself in the following way: If a 64 bit register is accessed that would result in a latency violation (e.g. it was the target of a 'hread' instruction but was subsequently accessed before the header load latency) , it crashed the assembler instead of reporting the violation gracefully. This is now resolved.
SDK-50337	692830	`88030_A0`	Packets arriving on the 1G ports were being redirected to incorrect queues due to incorrect PR ICC config mismatching. This has been fixed
SDK-50365	694983	`88650_A0 88650_B0`	Making sure bcm is attached before trying to detach it.
SDK-50368		`88750_A0 88650_A0 88640_A0`	Unused SOC properties (e.g. `policer_fairness_enable)` defined in config-sand.bcm were removed from this file

Table 118:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-50437	695853	88030_A0	There was a bug in the previous releases of the MDE that prevented correct parsing of variable length headers only for the first header (any variable length header that came afterwards has been parsing correctly all along; this is how IPv4 headers have been parsed for a long time). This bug has been fixed in the current release.
SDK-50440	695544	88030_A0	A summary CSV sheet is generated showing switch, key and port usage. To use either option: -suo "file-name" --output_summary_csv "file-name"
SDK-50441	695303	88030_A0	The ability to configure the ingress and egress queue parameters on a per queue basis is now supported by the MDE. In other words, it is possible to assign different parameters to each of the 64 ingress and 64 egress queues.
SDK-50442	695307	88030_A0	In the earlier releases of the MDE, the PPE property table did not get cleaned of the previous values consistently after a test has been run. This has been fixed.
SDK-50477	696358	88030_A0	Previous releases of the MDE had a bug with the following characteristics: If a masked 64 bit register (e.g of the form rr0[40:20]) was the destination of an 'hload' or 'hread' operation, the values read were put in starting at the lowest bits, i.e. the masking bits were ignored (SDK-50477). This has been fixed with this release.
SDK-50490	695720	88030_A0	Sync attribute has been added to CMU counter config. Valid values are "true" or "false".
SDK-50519	696880	88030_A0	Previous releases of the MDE had a bug where the TSR did not get updated for instructions in the egress task. This has been fixed.
SDK-50530		88650_A0 88660_A0	When setting FabricMC using Egress+Ingress MC, the OUTLIF in IRR_MCDB must be - '0'
SDK-50569	697394	88030_A0	Previous releases of the MDE implicitly limited the length of header fields to 32 bits (MAC fields were broken to 6 byte-length fields). This is now resolved.
SDK-50570	697442	88030_A0	Earlier releases of the MDE had a bug that prevented new direct-mapped tables to be added. This has been fixed in this release.
SDK-50571	697639	88030_A0	If the ingress/egress queues in the packet header get mangled (e.g. due to a microcode bug) the model now reports this.
SDK-50675		88650_B1	88550 and 88560 are Arad-SKU chips without Interlaken. During the Arad initialization, the 88550 & 88560 SKU were incorrectly considered TDM-only devices - fixed.
SDK-50718	699557	88030_A0	Ports are initialized in the following order: 6,13,7,14,12,11,10,2,3,4,5,8,9,0,1

Table 118:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.0***

SDK-50724	699541	88030_A0	To have two tables share the same memory the user must create two tables of the same width and size using the same ports and with the second table using the base address of the first table. For example:
			sample8 { table_capacity =(32 * 100)
			/*# "g3p1"."ocm"."Sample LRP OCM Port 8 Table." "Sample LRP OCM Port 8 Table. Test table." */
			index { test8i: 7 }
			entry { ocm_port(LRP_PORT_8, width=32) { pad:1 test8:31:0 } } }
			sample9 { table_capacity =(32 * 100) /*# "g3p1"."ocm"."Sample LRP OCM Port 9 Table." "Sample LRP OCM Port 9 Table. Test table." */
			index { test9i: 7 }
			entry { ocm_port(LRP_PORT_8, width=32, mem_base=sample8::base) { pad:1 test9:31:0 } } }

SDK-50748	699893	88030_A0	DM table results will be ordered correctly in the results registers.
SDK-50756		88650_A0 88650_B0 88650_B1 88660_A0	Added new diagnostics to display voq/vsq programmable counters: diag counter voq/vsq Queue=x (Interval=y) diag counter voq Basequeue=x (Interval=y)
SDK-50759		88650_A0 88650_B0 88650_B1 88660_A0	Added new diag "diag cosq voq id=<id> detail=1" to print given VOQ's attributes.
SDK-50779	696166	88650_A0	New APIs were added to dynamically enable/disable counter collection by counter processor engines: bcm_switch_service_get bcm_switch_service_set for more details about these APIs see Arad PP user manual (886X0-PG3XX)
SDK-50812	700562	88030_A0	The MDE now supports configuring the PPE variable in a similar manner as the property table entry. The configuration can be global or per-test. For example, to configure some fields in the ingress variable, enter the following within the ppe configuration (inside the scope of <ppe-m>... </ppe-m>: <ing_ppevar> <field-m> <name>mim_transit</name> <value>1</value> </field-m> <field-m> <name>lsp_gal</name> <value>1</value> </field-m> <field-m> <name>vrrp</name> <value>1</value> </field-m> </ing_ppevar>
SDK-50849		88750_A0 88650_A0 88650_B0 88650_B1 88660_A0	1. MBIST (internal memories BIST) is fixed to work on 88660. MBIST can be enabled at startup using the bist_enable soc property. 2. Starting with the 6.3.2 release, enabling of the mbist output is done using: dbm socdnx +mem +VERbose +err -cnt Instead of the previous: dbm soc +mem +VERbose +nor +err Using the reporting line above, all the memory debugging information previously available (for 88650 and for 88750) is displayed as before.
SDK-50894	701166	88030_A0	Release 144 of the MDE had an issue that the PPE header checker and LAG template must be specified, even if the application does not need it. Omitting these two optional parameters crashed the MDE. This has been fixed.

Table 118:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-50963		88650_A0 88650_B0 88660_A0	When using User-Header (e.g. in cascaded Ingress-Egress ACL or in VMAC), the user-header should be removed before the packet exits the system. This was not the case for OTMH program and Mac-in-Mac. This has been corrected.
SDK-50972	701844	88650_A0 88650ACP_A0 88650_B0 88650_B1	Table EGQ_FQP_NIF_PORT_MUX need to be tuned to avoid packet drops. Optimized internal table arrangement to prevent underrun and insure the desired ports rate. The fix is applied only on driver initialization.
SDK-51035		88650_A0 88650_B0 88650_B1	VLAN assignment according to port,protocol: VLAN assignment procedure is according to profile. Increased the number of port protocol entries per profile from 10 to 16
SDK-51048	700857	56850_A0 56850_A1 56850_A2	PFM_RULE_APPLY field in IGMP_MLD_PKT_CONTROL register cannot be controlled in previous release. Added support for controlling this bit by pkt protocol control approach.
SDK-51093	705776	88030_A0	The condition (header access latency) was considered cleared after one cycle (i.e. next instruction) rather than two cycles.
SDK-51170		88660_A0	OAM: Support RDI generation method. Generation method is configured through the bcm_oam_endpoint_create api with the following flags2: BCM_OAM_ENDPOINT2_RDI_FROM_RX_DISABLE /* RDI bit on outgoing packets may be taken from RDI indication on received packets. */ BCM_OAM_ENDPOINT2_RDI_FROM_LOC_DISABLE /* RDI bit on outgoing packets may be taken from LOC indication of peer endpoint. */
SDK-51184	705114	88030_A0	The LUG is out of date with respect to the COP load latency, the correct value is 40.
SDK-51351	709776	88030_A0	I can see the code the load latency is changed from 37 to 40
SDK-51368	707551	56830_A1 56830_A0 56830_A2	BCM56830 is considered as a switch instead of a fabric and attached with proper drivers. SDK implementation has been corrected based on this determination
SDK-51498	696152	88130_B0	QE2000 to Sirius traffic issue was resolved with a fix to bcm_fabric_crossbar_connection_set() to set up both A and B plane connections to support plane crossover.
SDK-51528	711580	All	Fixed the issue with packet drop counter when the packet is dropped by policer.
SDK-51541 SDK-50704		88650_A0	In order to detect and fix ECC2 and parity errors, one can use the BCM switch control bcmSwitchCacheTableUpdateAll. The procedure will go over all cached memories, read them from HW, and in case it detects an error, a matching interrupt will be initiated to be corrected by the appropriate corrective action. When caching memories, it is recommended to update all cached memories before initiating a WB/ISSU cycle. The cached memories are read from the HW during WB/ISSU. Updating all cached memories ensures that all potential errors are handled using the available shadow data.

Table 118:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.0***
SDK-51643		56340_A0	Fixed and tested on GTO/BCM56340A. BCM init and rc failures aren't happening anymore.
SDK-51689	713650	88650_B1	In BCM8865X, a bug at egress HW was mishandling packets that being terminated to size of 192-255 Bytes. In BCM88660, this HW bug was fixed. Enabling this bugfix during the Driver init is inserted.
SDK-51821	716070	88030_A0	Added check for invalid combination (Simple64 & Automatic mode): Error! [87509] null->0:0->0.1 = Counters group ertctr: Simple64 counters don't support automatic mode.
SDK-51827	716807	88030_A0	Resolves ucode reload issue seen in 2_146 and TOT.
SDK-51881	702602	56640_B0 56850_A2	Vlan Service queuing bugs addressed. 1. gport_attach/detach to take care of internally attaching the given number of Queues during add. 2. Queue alignment of Vlan queuing changed to 1(no alignment required).
SDK-52216	711504	56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56820_A0 56820_B0 56800_A0 56746_A0 56745_A0 56744_A0 56743_A0 56740_A0 56725_A0 56720_A0 56700_A0 56689_B0 56685_B0 56685_A0	A request was made to add the ability to override "protocol" field in SKB before pushing packet into network stack. the following fields were added in the packet filter structure to support this request. : int dest_proto; /* If non-zero this value overrides the default protocol type when matching packet is passed to network stack. */ int mirror_proto; /* If non-zero this value overrides the default protocol type when matching packet is passed to network stack. */ add corresponding fields in packet filter structure to configure the desired protocol type.
SDK-52355		56850_A0 56850_A1 56850_A2	Support has been added for retry in mem insert and delete for hash tables. Inline hash memory recovery was implemented for insert and delete operations.When an insert/delete operation encounters a parity error, the inline recovery routine will be invoked. The inline recovery routine will calculate different hash buckets in different hash memory banks based on the entry that will be inserted/deleted, then restore the each bucket in these banks. For new-added hash key types in Trident2 hash tables, support for these key types in hash entry comparing routine has also been added.
SDK-52381	717920	56850_A0	In earlier releases, L3 Conflict Get, bcm_td2_l3_conflict_get() was broken. This has been resolved.
SDK-52443	705504	88650_A0	Case Summary: Traffic drops at ingress on a newly added LAG member, if it is the first member on the ARAD device after cross connection created. To avoid the problem, API support was added. Using the following function flags bcm_trunk_member_addbcm_trunk_member_delete and bcm_trunk_set the user can update only egress or ingress tables. The user can update only egress tables, configure relevant port parameter and then update ingress port, with this sequence there will be no traffic drop.
SDK-52448	723913	56450_A0	HQOS support is added for UNI ports on Katana2
SDK-52471	723924	88030_A0	Order issue addressed in template generated code.

Table 118:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-52514	725210	88030_A0	Release 149 had a bug where the MDE crashed if a hash template was not configured for the ingress queue. This has been addressed in this release.
SDK-52521	724174	56850_A0	In the previous release, in function `_soc_td2_alloc_sched()`, HQOS hierarchy was being assumed. If users did not use the same hierarchy as defined in `_td2_port_lls_config()`, issues would be seen. In this release, a LLS port doesn't clear other ports' hardware resource when `bcm_cosq_gport_add()` is called on Trident2 chips.
SDK-52722		88650_A0	`bcm_l3_ingress_create` now returns an error if the flag `BCM_L3_INGRESS_WITH_ID` is not enabled (instead of just ignore)
SDK-52758	727046	88030_A0	BCM88030: fixed bug where MPLS label was overwriting the IP address RCE key field.
SDK-52767		88750_A0 88650_A0 88750_B0 88650_B0 88650_B1 88660_A0	"show features" diagnostics was added.
SDK-52895	729741	All	RPC has been enabled for the HASH bank APIs.
SDK-52970	730058	All	L2 `matched_traverse` used a loop to test availability of MOD FIFO, and if MOD FIFO became hung, the loop became endless and eventually caused MOD FIFO thread to become dead. Added a timeout to break out of the loop if MOD FIFO hangs.
SDK-53012	715940	88030_A0	Fixed the wrong action type for mirror & drop in egress RCE action table.
SDK-53021	720668	56850_A0	Updated the documentation related to `BCM_PORT_CONGESTION_CONFIG_DESTMOD_FLOW_CONTROL`
SDK-53070	688151	56850_A2	Two command options are added for the eye margin functional calls. The syntax example is phy diag xe0 veye lane=0xc `time_upper_bound=16` The "lane" option specifies which lane in a given port is enabled for the eye measurement. If the lane is not specified, the default is 0 which means all the lanes are enabled for the eye measurement in a given port. 0xC means lane 2 and lane 3 of the port is enabled. Each binary bit of the value represents a lane. The `"time_upper_bound"` is to specify the total max time limit for a given eye measurement node. Its unit is second. The default is 256 seconds. Note that this option only accepts the value equal to or larger than 4 seconds.
SDK-53077	731557	88030_A0	For IPv4, we now could use `max_capacity_limit/240` to estimate the number of tcam entries need to be used. For IPv6, use `max_capacity_limit/168` to estimate it.

Table 118:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-53112	677748	88030_A0	As of the current release of the MDE, the TMU MAC table subkeys can be split into three (as opposed to two: VSI and MAC) fields ONLY for testing on the model as follows: An optional 1-bit field, called 'bmac' can be specified as part of the MAC subkey. This means that the MAC subkey can be optionally split into three fields (1 bit BMAC, 15 bit VSI and 48 bit MAC address) This is meant to used in simulating a PBB (Mac-in-Mac) header where the bmac field can be 1 to specify that the MAC address in the subkey is a bridging MAC or 0 to specify that it is a customer (inner) MAC. Since this field is optional, the default value is 0 which covers both the cases of a non-PBB header as well as the inner MAC of a PBB header. The UcTst.xml file that is provided with this release has placeholder examples of this new usage.
SDK-53115	731716	56850_A0 56850_A1 56850_A2	For TCAM memories protected by SER engine, corrupt bitmaps have been added to track SER errors detected on them. SER correction logic will filter duplicated SER errors via this corrupt bitmap.
SDK-53127	730044	56334_B0 56334_A0	In an earlier release switching double tagged frames between layer 2 logical ports on Enduro was inconsistent with TR3/TR2 behavior. This has been correction by synchronizing the behavior of double tagged frames switching on Enduro with TR2's behavior.
SDK-53198	733029	56640_A0 56440_A0 56641_A0 56450_A0	The problem in existing code was - `bcm_l3_egress_get()` was not able get the `mpls_qos_map_id` i.e logical qos id (if object was created by `bcm_qos_map_create).` In this release to solve this issue, a new routine `_egr_qos_hw_idx2id()` has been created. This converts the hardware index to logical qos id. This function is used in the `bcm_l3_egress_get` to retrieve the `mpls_qos_map_id.` This function can handle both the `qos_id` created by `bcm_qos_map_create()` and `bcm_mpls_exp_map_create().`

Table 118:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-53204	732754	88030_A0	Certain configuration parameters are required to set and configure the ILKN OOB flow control. The following illustrates how the required parameters are used and when to use them Some parameters are specific to Caladan3 chip. 1) Enable Interlaken Flow control when there is an interlaken port • fc_oob_type_<ilx>=2 2) Enable Interlaken flow control when there is no interlaken port • fc_type_il_line=1 • fc_type_il_fabric=1 3) Default Calendar length is 64, if not the following has to be set appropriately • fc_calendar_length_il_line=<len> • fc_calendar_length_il_fabric=<len > 4) Debugging 1.Ignore the FC OOB status • ilkn_interface_status_oob_ignore= 1 2.Enable Loopback of FC data • fc_oob_loopback_<ilx> = 1
SDK-53218	727679	88650_A0 88650_B0 88650_B1	Port TPIDs: When deleting TPID to default behavior with API bcm_port_tpid_delete or bcm_port_tpid_delete_all , TPID profile wasn't changed correctly.
SDK-53311	733395	56850_A2	Operations in soc_l2x_freeze/thaw() for TD2 have been optimized by using ING_MISC_CFG2 CML_NEW_OVERRIDE/ CML_MOVE_OVERRIDE to disable/enable the learning instead of modifying individual port/svp table entries.
SDK-53346	731211	88650_A0	The member ID of Trunk port as it written to the IHP_PTC_SYS_PORT_CONFIG & IHP_VIRTUAL_PORT_TABLE was not the index in the IRR lag mapping table. We fixed lag member delete/add to keep this tables synced.
SDK-53348	733779	88030_A0	In the previous release an issue was discovered with clearing Interlaken counters. This has been fixed. When using "clear counters", we now clear those counts for both hardware counters and the software variable, then we will reset those counts.

Table 118:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-53367	735381	88030_A0	SDK-53367 PTN6500 - line rate test cause "ped egress drop" issue. This is actually a COP access constraint that was not covered. The constraint is the 2nd rule of the following: 1st rule: Independent of the targeted instance, COP accesses always have AT LEAST a 64 instruction resource shadow. (covered by 51017 R-000-10 COP Resource Shadow (load) and 51018 R-001-10 COP Load Latency) 2nd rule: Per COP instance, the assembler maintains a counter (initialized to 0). The counter value must be <= 128 in order to access the associated COP instance without violating the constraint. The counter is maintained as follows: 1.) Starts at 0. 2.) Add 128 to the counter when the associated COP instance is accessed. 3.) For each instruction slot, if the counter is non-zero, and it has been at least 64 instructions since the associated COP instance has been accessed, subtract 2 from the counter. Note that this is not the same as the COP port resource shadow (see 1st rule) which is common to both COP instances. New resource constraint: R-002-10 New error number: 51064 Example error message: Error! [51064] copTest.lrp3->90:5->1.76 => 223:5->1.199 = constraint R-002-10 COP per instance access constraint violation. shadow:1 Next safe instruction for COP instance 0 access is: 200
SDK-53370	726683	88650_B1	When using FCoE example CINT, the FCoE header in the forwarded packet was omitted. The trap ID that is used for FCF workaround, which fixes the forwarding header offset was wrong and is now fixed.
SDK-53385	721111	88650_A0	In RX snoop, the number of HW snoop commands is 16, where 0 is reserved for packets which are not snooped. Due to a SW bug, the number of available snoop commands was 15 and not 16. This is fixed. This fix was reverted in 6.3.5 because it breaks ISSU and can be taken from TOT as a patch.
SDK-53414	734150	56850_A0	In the previous release we did not support HG13 on TD2. In this release support has been added for speed 13000M. Additionally in soc_td2_port_asf_speed_set(),if speed = 0xe, speed 13000M duplex full will be selected.
SDK-53449	733944	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1 56851P_A1 56851_A1 56850_A2 56851_A2 56851P_A2 56854_A2 56853_A2 56852_A2 56855_A2 56851_A0 56852_A0 56852_A1 56853_A0 56853_A1	In the previous release. bcmportControlDoNotCheckVlan was being overwritten by unrelated port API calls. This has been fixed.

Table 118:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.0*
SDK-53453	675993	`56846_A0 56845_B0 56840_A0 56640_A0 56440_A0 56850_A0 56855_A0 56843_B0 56340_A0 56640_B0 56440_B0 56850_A1 56850_A2 56344_A0 56342_A0`	Added support for MIM payload tpid select and MIM hash by using payload or tunnel header.
SDK-53488	736297	`88650_A0`	ARAD does not support Type-4 VCCV (GAL over PWE). We propose a solution to trap GALoPWEoLSPoETH packets to CPU by using bcmRxTrapMplsUnexpectedNoBos trap. Field processor is used to change the MPLS InLif to PWE Inlif, so the trapped packet contains PWE InLif in the PPH. NOTE: In ARAD soc property `custom_feature_mpls_termination_check_bos_disable` should be set. In ARAD+ no soc property is required. For usage example see `cint_gal_o_pwe_o_mpls.c`
SDK-53515	734789	`5615_A0`	HR2: QSGMII running as SGMII mode was showing the wrong duplex attribute. In this release we have fixed the duplex get function for qsgmii serdes in sgmii mode
SDK-53558	716344	`88030_A0`	The exception byte counter was not incremented when a packet is dropped due to drop tag (or) drop untag configuration. This is fixed.
SDK-53560	719326	`88030_A0`	The pvv2e.hit bit handling is fixed. The `soc_sbx_g3p1_utils_pvv2e_update()` & `soc_sbx_g3p1_utils_pvv2e_add()` functions sets the hit bit by default.
SDK-53563	736727	`56334_B0 56334_A0`	Fixed error return value of `bcm_mpls_label_s?at_get/get32` on Enduro
SDK-53612	728198	`88650_B1`	When working in 2P or 1P mode, ISQ root shaper doesn't work correctly (traffic is not shaped regardless shaper configuration).
SDK-53613	735136	`88650_A0 88650ACP_A0 88650_B0 88650_B1`	When setting WRED using `bcm_cosq_gport_discard_set,` and using min/max threshold values close to the limit of 256MB-1, an error was returned. This was fixed, and the range of the min/max WRED thresholds was extended up to 2GB. The actual value that can be specified is up to 2GB-1 sue to the range of the `int` structure field that specifies it.

Table 118:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-53639	737816	All	bcmFieldQualifyL3Ingress qualifier offsets are updated for Ingress Field Processor to match with regfile (56850). Problem : Previously the qualifier set was showing "Feature Unavailable" error during group create. This was due to missing initialization of L3Ingress qualifier. Solution: With this fix the group create will cause "No resources for operation" error for the qualifier set mentioned above. This is because after adding bcmFieldQualifyL3Ingress to the Groups QSET the KEY width is exceeding what TD2 IFP H/W can support. Customer has to remove either bcmFieldQualifyInterfaceClassL3 or bcmFieldQualifyIntPriority qualifier from the Groups QSET set to add bcmFieldQualifyL3Ingress to existing Group. OR Customer has to create a new Field Group with bcmFieldQualifyL3Ingress qualifier in it.
SDK-53640		`56334_B0 56334_A0`	In earlier releases a crash was introduced when initializing BCM56634 via changes added in `soc_do_init`. In this release we have added device checking for the new block of code introduced to change the PCIe SerDes deemphasis on certain devices (fix for SDK-50513).
SDK-53650		`88650_A0 88650_B0 88650_B1`	Fixed the crash `in_bcm_dpp_rx_packet_parse` when called with `BCM_ARAD_PARSE_PACKET_IN_INTERRUPT_CONTEXT`. Registered are not accessed when working in interrupt context.
SDK-53654		`88650_A0 88650_B0 88650_B1 88660_A0`	Fixed "diag rates sch" shell command crash which is caused by reading non-existent register.
SDK-53675		`88650_A0 88650_B0 88660_A0`	BFD packets may now be trapped using pre-defined traps. When calling `bcm_bfd_endpoint_create()`, the `remote_gport` field may be set to a valid gport for trapping BFD frames to that gport, `GPORT_INVALID` for the default behavior or `remote_gport` may be set to a pre-configured trap code. For the latter, call `bcm_rx_trap_type_create()` to get a trap code, `bcm_rx_trap_set()` to set the trap code with a valid `dest_port` configured in `bcm_rx_trap_config_t`, `BCM_RX_TRAP_UPDATE_DEST` and `BCM_RX_TRAP_TRAP` flags set. Then set `remote_gport` to the said trap code before calling `bcm_bfd_endpoint_create()`.
SDK-53731	739297	`88750_A0 88750_B0`	"diag queues" command shell wasn't functional over dual pipe.
SDK-53767		`88650_A0`	cleaned HW access that were causing error prints during warm reboot (due to statistic threads that would perform HW access)

Table 118:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.0*
SDK-53824		`56450_A0`	In previous releases a crash could occur with subport configuration. This has been addressed by correcting the wrong assumption of COE subport configuration for calculating `op_nodes` for physical ports. Now it is purely based on number of `op_nodes` consumed by each physical port in sequence (CPU,LPBK,1..40)
SDK-53826		`88660_A0`	PON: `bcm_vlan_port_create` set incorrect configuration when having 3 tags manipulation under `bcm886xx_vlan_translate_mode=1.`
SDK-53837		`88650_B0 88650_B1`	Fix documentation of `cint_vswitch_cross_connect_p2p.c` to load all the cints in correct order.
SDK-53839		`88650_B0 88650_B1`	VPLS: Added cint `cint_vswitch_vpls.c` support in index mpls mode that enables termination of up to 3 labels. Index mode is set using soc property `mpls_termination_label_index_enable.`
SDK-53867	740320	`56850_A0 56850_A1 56850_A2`	One of the following solutions can be used to address the persistent link flap problem with CR4 + AutonegOn on ports: (a) Do NOT enable `RX_SERDES_LOS` and Fast linkscan property in the configuration. This means to disable the SOC property `rx_serdes_los,` or, EXCLUDE the port(s) from the SOC property `rx_fast_los_link_{port}.` (b) If the user wants to be able to remove/add ports into fast linkscan dynamically,user can now disable the fast linkscan port control "bcmPortControlRxFastLOS" : `bcm_port_control_set/get(unit,` port, bcmPortControlRxFastLOS, ...).
SDK-53891		`88650_A0`	Relevant only for TDM bypass mode: Warm boot would reset some of the TDM fabric direct routing configuration, and cause later configuration of it to be incorrect.
SDK-53946		`88650_B1 88660_A0`	Important note: in Fiber channel APIs, due to an API change, the user must replace `bcm_fcoe_zone_entry_t->vsan.vsan` by `bcm_fcoe_zone_entry_t->vsan_id,` e.g. in `bcm_fcoe_zone_add` API.
SDK-54096		`88650_A0`	The private header that includes the packet size was supported in previous version, but collide with other PP features that are supported by the egress editor. This fix resolves the issues when trying to enable multiple PP features while still maintaining the usage of the "size header" addition on top of the packet header.
SDK-54378		`88650_A0`	To debug more easily warmboot issues, a SW state dump is available via BCM>diag ssdump The SW state dump output to screen can now be disabled.

Table 118:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-56158	756172	56440_A0	Problem description: When L2 MAC Table is full, customer was unable to create the BFD session due to MAC table full issue. Fix description: This issue is fixed by calling _bcm_l2_hash_dynamic_replace function, if soc_mem_insert function returns BCM_E_FULL. and also code changes are done in _bcm_l2_hash_dynamic_replace function to support BFD_KEY for Katana device.