Software Development Kit
Release Notes
SDK 6.4.1

June 26, 2014

Broadcom
Network Switching

Section 1: About This Document

These are the Release Notes for the Broadcom Network Switching Software Development Kit, Release 6.4.1.

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-PG641-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

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. This release contains Early Access support for BCM534XX A0 device in external and internal host modes. For internal host mode, the SDK for BCM5341X series family shall be compatible with LDK version 3.5.x or later

The supported devices with respective port configurations are as listed: (x denotes as 0/1)

- BCM534x2: 8 x 1G/2.5G/5G/10G
- BCM534x8: 24 x 1G/2.5G
- BCM534x1 option1: 4 x QSGMII, 8 x 1GbE, 4 x 10GbE
- BCM534x1 option2: 4 x QSGMII, 8 x 1GbE, 2 x 10Gbe, 2 x HiGigDuo[13]
- BCM534x5: 16 x 1G/2.5G/5G/10G
- BCM534x6: 12 x 10GbE, 8 x 2.5GbE, 4 x 5GbE/2.5GbE

Each device has passed in-house diagnostic and traffic tests with loopback modes of MAC, internal PHY and external PHY, including

- 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
- L3 test
- SER test
- PCI Compliance and S-Channel test

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 channel variations

RANGER2

BCM5606X 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. This release contains Early Access support for BCM5606X A0 device in external and internal host modes. For internal host mode, the SDK for BCM5606X series family shall be compatible with LDK version 3.5.x or later

The supported devices with respective port configurations are as listed:

- BCM56060: 16 x 1G/2.5G/5G/10G
- BCM56063: 4 x QSGMII, 4 x 10GbE (XFI)
- BCM56064 option1: 4 x QSGMII, 8 x 1GbE, 4 x 10GbE
- BCM56064 option2: 4 x QSGMII, 8 x 1GbE, 2 x 10Gbe, 2 x HiGigDuo[13]

Each device has passed in-house diagnostic and traffic tests with loopback modes of MAC, internal PHY and external PHY, including

- 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
- L3 test
- SER test
- PCI Compliance and S-Channel test

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 channel variations

BCM88950 (FE3200) PREVIEW RELEASE

The Broadcom BCM88950 is the fourth generation in the Dune product line of Fabric Element (FE) devices - following BCM88750 (FE1600)

BCM88950 is a self-routing cell-based switching element and enables BCM88670 switch-based systems to build flexible networking platforms providing user bandwidth over 960 Tbps.

This is a preview release of BCM88950 driver. It is provided for reference, and allows an early integration of the customer's application with the SDK APIs that are currently supported. The list of APIs which will be supported for BCM88950, and the ones that are already implemented, can be found at the SDK-6.4.1 support matrix document.

The following BCM88950 exclusive features were added compared to the 6.4.1 EA2 version

- BCM88950 Multistage exclusive features: Asymmetrical FE13, FE13 local routing

BCM88670 (JERICHO) PREVIEW RELEASE

The Broadcom 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 wirespeed 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.

This is a preview release of BCM88670 driver. It is provided for reference, and allows an early integration of the customer application with the SDK APIs that are currently supported. The list of APIs which will be supported for BCM88670, and the ones that are already implemented, can be found at the SDK-6.4.1 support matrix document.

For differences in the BCM SDK SW interface between BCM88660 and BCM88670 devices, please refer to BCM88670 and BCM88660 SW compatibility document.

BCM88670 preview driver supports the following features:

- Registers access
- PCID simulation

- Access TR tests
- Initialization sequence
- API definition and initial implementation of UC and MC data plane
- Link statistics

Arad+ features implementation, including

- L2
- STG
- VLAN
- Consistent hashing
- Field Processor
- MIM

BCM88650 (ARAD) AND BCM88660 (ARAD+) RELEASE

Several enhancements and changes were done on existed features between SDK SW between 6.4.X releases and 6.3.X release. Those changes require application level changes in order to move existed application from 6.3.X release to 6.4.X release. For all important changes and clarifications, please refer to BCM88660 SW Compatiblity between 6.4.X and 6.3.X BCM SDK document.

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 1.67 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 1.06 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 2: Validated Warmboot upgrades

Software upgrade	*Supported*
6.4.0 to 6.4.1	Yes
6.3.7 to 6.4.1	Yes

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

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

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 3: Supported Switch Devices

Family	Devices	Description
BCM53400	BCM53401 A0	24-port GbE plus 4-port 10GbE Multilayer Ethernet Switch with HiGi Uplinks
	BCM53402 A0	8-port 10GbE Multilayer Ethernet Switch
	BCM53405 A0	16-port 10GbE Multilayer Ethernet Switch
	BCM53406 A0	12-port 10GbE plus 8-port 2.5GbE and 4-port 5GbE/2.5GbE Multilayer Ethernet Switch
	BCM53408 A0	24-port 1GbE/2.5GbE Multilayer Ethernet Switch
	BCM53411 A0	24-port GbE plus 4-port 10GbE Multilayer Ethernet Switch with HiGi Uplinks and integrated CPU
	BCM53412 A0	8-port 10GbE Multilayer Ethernet Switch with integrated CPU
	BCM53415 A0	16-port 10GbE Multilayer Ethernet Switch with integrated CPU
	BCM53416 A0	12-port 10GbE plus 8-port 2.5GbE and 4-port 5GbE/2.5GbE Ethernet Switch with integrated CPU
	BCM53418 A0	24-port 1GbE/2.5GbE Multilayer Ethernet Switch with integrated CPU
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
BCM56450	BCM56248L B0	11xGE + 8x2.5G
BCM56846	BCM56831 A1	Trident+ SKU - 24-port 10GE switch with 40GE support for embedded applications

Table 4: Preview Switch Devices

Family	Devices	Description
BCM56450	BCM56455	Katana2 Access 1 x XAUI + 8 x GE without L3 routing and MPLS features
BCM56450	BCM56248L w/10G Upgrade	8xGE+8x10G with reduced table sizes
DNX	BCM88670_A0	720 Integrated Packet Processor, Traffic Manager, and Fabric Interface Single-Chip Device
DNX	BCM88950_A0	3.2 Tbps self-routing switching element
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
BCM56850	BCM56838 A2	Trident2 SKU - Ready for Bringup - 72/320G Devices with 1.25/3.125/6.25G Serdes and 4 SFIs
BCM56846	BCM56835C A1	Trident+ SKU - (64 x 10 GbE) + (4 x 1 GbE)
BCM56846	BCM56849 A1	Trident+ SKU - (56 x 1GbE/2.5GbE) + (8 x 10GbE)

Table 4: Preview Switch Devices

Family	Devices	Description
BCM56850	BCM56834 A2	Trident2 SKU - High density 10G and 40G switch for embedded applications
BCM56640	BCM56044 B0	Ranger+ SKU - 100G + 3xF.HG[42] + 1GE
BCM56846	BCM56847 A1	Trident+ SKU - (64 x 10 GbE) + (4 x 1 GbE)
BCM56640	BCM56545K	Triumph 3 SKU - Device recognition only- 48-port GE switch + 4x10GE + 4xHG[42] / 40GE
BCM56640	BCM56546K	Triumph 3 SKU - Device recognition only- 28-port GE switch + 4x10GE + 4xHG[42] / 40GE
BCM56240	BCM56245	Saber SKU -2x (10GbE/4x 1GbE/4x 2.5GbE) + 2x 10GbE/12GbE/13GbE, IEEE 1588 enable
BCM56242	BCM56246	Saber SKU -10x 1GbE/2.5GbE, IEEE 1588 enabled
BCM56150	BCM53347	Wolfhound SKU - 24-port GbE Multilayer WebSmart Switch with 6xQSGMII + 4x1/10G

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)

Table 6: Preview PHYS

Device	Driver Family	Description
BCM82328_B0	82328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version 7 "(Preview)
BCM54220SE	54220	Dual Copper/Fiber Gigabit Ethernet Transceiver (Bringup) EEE , SyncE and 1588 not yet supported
BCM82322_B0	82328	12port Gallardo28 supporting 12x10G, 3x40G, 1x100G;

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-PG641-R. (See section 2 earlier in this document for availability).

DEBUG SUBSYSTEM

This version of the SDK uses a new logging framework called Broadcom System Logging (BSL) which is independent of the Configuration Manager (CM.) This change affects the system initialization, and SDK-based applications will require the following changes in the SDK system interface:

- The function `soc_cm_init()` no longer takes any parameters.
- A new function `bsl_init()` must be called to initialize the BSL framework.

The SDK source code contains new files which demonstrates how application-provided BSL hooks can be implemented. Please see the SDK Platform Guide for updated documentation on the CM and the BSL framework.

BCM FUNDAMENTALS

BCM MODULES

A new module `BCM_MODULE_UDF` 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 8: 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

Table 8: BCM Module IDs

BCM Module ID	***String Equivalent***
BCM_MODULE_MPLS	mpls
BCM_MODULE_FABRIC	fabric
BCM_MODULE_MIM	mim
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

BIDIRECTIONAL FORWARDING DETECTION

A new BFD Tunnel type, defining the encapsulation type for the session, has been added.

Table 9: BFD Tunnel Types

Flag	***Description***
bcmBFDTunnelTypePweGal	MPLS PWE with GAL, GACH

A new BFD Endpoint flag has been added.

Table 10: BFD Endpoint Info Structure Flag Definitions

Flag	***Description***
BCM_BFD_ENDPOINT_REMOTE_UPDATE_STATE_DISABLE	Disable updating RMEP DB.

The new sampling_ratio, loc_clear_threshold and ip_subnet_length fields in data structure bcm_bfd_endpoint_info_t have been added.

```
/* BFD endpoint object. */
typedef struct bcm_bfd_endpoint_info_s {
    ...
    int sampling_ratio;         /* percentage of packets sampled to the
```

```
                                * CPU
                                */
    uint8 loc_clear_threshold; /* Number of packets required to reset
                * the Loss-of-Continuity status per
            * endpoint.
                                */
    uint32 ip_subnet_length;   /* The subnet length for incoming packet
                * validity check. Value 0 indicates no
            * check is performed, positive values
            * indicate the amount of MSBs to be
            * compared.
                                */
} bcm_bfd_endpoint_info_t;
```

REGEX API

A new API bcm_regex_session_delete_all() has been added in this release.

bcm_regex_session_delete_all

Deletes all the entries from the regex session table.

Syntax

```
#include <bcm/bregex.h>
int bcm_regex_session_delete_all(int unit);
```

Parameters

unit	(IN) Unit number.

Description

Deletes all the entries (IPv4 and IPv6) from the regex session table.

Returns

BCM_E_xxx

CLASS OF SERVICE QUEUE CONFIGURATION

New CoSQ Control Types have been added.

Table 11: CoSQ Control Type Values

Value	Description	Arg value
bcmCosqControlEgressPortPoolYellowLimitBytes	Egress port service pool limit setting for yellow packets.	Shared port service pool size for yellow packets.
bcmCosqControlEgressPortPoolRedLimitBytes	Egress port service pool limit setting for red packets.	Shared port service pool size for red packets.

New cosq gport types have been added for `bcm_cosq_gport_handle_get()` API.

Table 12: gport types

bcmCosqGportTypeFabricPipe	***Fabric all pipes gport***
bcmCosqGportTypeFabricPipeEgress	Fabric egress pipes gport
bcmCosqGportTypeFabricPipeIngress	Fabric ingress pipes gport
bcmCosqGportTypeFabricPipeMiddle	Fabric middle pipes gport
bcmCosqGportTypeGlobalFabricMeshCommonLocal0	mesh common local 0
bcmCosqGportTypeGlobalFabricMeshCommonLocal1	mesh common local 1
bcmCosqGportTypeGlobalFabricMeshCommonDev1	mesh common device 1
bcmCosqGportTypeGlobalFabricMeshCommonDev2	mesh common device 2
bcmCosqGportTypeGlobalFabricMeshCommonDev3	mesh common device 3
bcmCosqGportTypeGlobalFabricMeshCommonMc	mesh common multicast
bcmCosqGportTypeGlobalFabricClosCommonLocal0	clos common local 0
bcmCosqGportTypeGlobalFabricClosCommonLocal1	clos common local 1
bcmCosqGportTypeGlobalFabricClosCommonFabric	clos common fabric
bcmCosqGportTypeGlobalFabricClosCommonUnicastFabric	clos common unicast fabric
bcmCosqGportTypeGlobalFabricClosCommonMulticastFabric	clos common multicast fabric
bcmCosqGportTypeGlobalFabricMeshLocal0OcbHigh	mesh local 0 ocb high
bcmCosqGportTypeGlobalFabricMeshLocal0OcbLow	mesh local 0 ocb low
bcmCosqGportTypeGlobalFabricMeshLocal0MixHigh	mesh local 0 mix high
bcmCosqGportTypeGlobalFabricMeshLocal0MixLow	mesh local 0 mix low
bcmCosqGportTypeGlobalFabricMeshLocal1OcbHigh	mesh local 1 ocb high
bcmCosqGportTypeGlobalFabricMeshLocal1OcbLow	mesh local 1 ocb low
bcmCosqGportTypeGlobalFabricMeshLocal1MixHigh	mesh local 1 mix high
bcmCosqGportTypeGlobalFabricMeshLocal1MixLow	mesh local 1 mix low
bcmCosqGportTypeGlobalFabricMeshDevlOcbHigh	mesh device 1 ocb high
bcmCosqGportTypeGlobalFabricMeshDev1OcbLow	mesh device 1 ocb low
bcmCosqGportTypeGlobalFabricMeshDev1MixHigh	mesh device 1 mix high
bcmCosqGportTypeGlobalFabricMeshDev1MixLow	mesh device 1 mix low
bcmCosqGportTypeGlobalFabricMeshDev2OcbHigh	mesh device 2 ocb high
bcmCosqGportTypeGlobalFabricMeshDev2OcbLow	mesh device 2 ocb low
bcmCosqGportTypeGlobalFabricMeshDev2MixHigh	mesh device 2 mix high
bcmCosqGportTypeGlobalFabricMeshDev2MixLow	mesh device 2 mix low
bcmCosqGportTypeGlobalFabricMeshDev3OcbHigh	mesh device 3 ocb high
bcmCosqGportTypeGlobalFabricMeshDev3OcbLow	mesh device 3 ocb low
bcmCosqGportTypeGlobalFabricMeshDev3MixHigh	mesh device 3 mix high
bcmCosqGportTypeGlobalFabricMeshDev3MixLow	mesh device 3 mix low
bcmCosqGportTypeGlobalFabricMeshMcOcbHigh	mesh multicast ocb high
bcmCosqGportTypeGlobalFabricMeshMcOcbLow	mesh multicast ocb low
bcmCosqGportTypeGlobalFabricMeshMcMixHigh	mesh multicast mix high
bcmCosqGportTypeGlobalFabricMeshMcMixLow	mesh multicast mix low
bcmCosqGportTypeGlobalFabricMeshLocal0	mesh local 0
bcmCosqGportTypeGlobalFabricMeshLocal1	mesh local 1
bcmCosqGportTypeGlobalFabricMeshMc	mesh multicast
bcmCosqGportTypeGlobalFabricMesh	mesh
bcmCosqGportTypeGlobalFabricClosLocal0OcbHigh	clos local 0 ocb high
bcmCosqGportTypeGlobalFabricClosLocal0OcbLow	clos local 0 ocb low
bcmCosqGportTypeGlobalFabricClosLocal0MixHigh	clos local 0 mix high
bcmCosqGportTypeGlobalFabricClosLocal0MixLow	clos local 0 mix low

Table 12: gport types

bcmCosqGportTypeFabricPipe	***Fabric all pipes gport***
bcmCosqGportTypeGlobalFabricClosLocal1OcbHigh	clos local 1 ocb high
bcmCosqGportTypeGlobalFabricClosLocal1OcbLow	clos local 1 ocb low
bcmCosqGportTypeGlobalFabricClosLocal1MixHigh	clos local 1 mix high
bcmCosqGportTypeGlobalFabricClosLocal1MixLow	clos local 1 mix low
bcmCosqGportTypeGlobalFabricClosUnicastFabricOcbHigh	clos fabric unicast ocb high
bcmCosqGportTypeGlobalFabricClosUnicastFabricOcbLow	clos fabric unicast mix high
bcmCosqGportTypeGlobalFabricClosUnicastFabricMixHigh	clos fabric unicast ocb low
bcmCosqGportTypeGlobalFabricClosUnicastFabricMixLow	clos fabric unicast mix low
bcmCosqGportTypeGlobalFabricClosFmqGuaranteedOcb	clos fabric multicast guaranteed ocb
bcmCosqGportTypeGlobalFabricClosFmqBestEffortOcb	clos fabric multicast best effort ocb
bcmCosqGportTypeGlobalFabricClosFmqGuaranteedMix	clos fabric multicast guaranteed mix
bcmCosqGportTypeGlobalFabricClosFmqBestEffortMix	clos fabric multicast best effort mix
bcmCosqGportTypeGlobalFabricClosLocal0	clos local 0
bcmCosqGportTypeGlobalFabricClosLocal1	clos local 1
bcmCosqGportTypeGlobalFabricClosLocal0High	clos local 0 high
bcmCosqGportTypeGlobalFabricClosLocal0Low	clos local 0 low
bcmCosqGportTypeGlobalFabricClosLocal1High	clos local 1 high
bcmCosqGportTypeGlobalFabricClosLocal1Low	clos local 1 low
bcmCosqGportTypeGlobalFabricClosOcbHigh	clos ocb high
bcmCosqGportTypeGlobalFabricClosMixHigh	clos mix high
bcmCosqGportTypeGlobalFabricClosOcbLow	clos ocb low
bcmCosqGportTypeGlobalFabricClosMixLow	clos mix low
bcmCosqGportTypeGlobalFabricClos	clos

New flags for new BCM CoSQ gport have been added.

Table 13: BCM CoSQ gport flags

Flags	***Notes***
`BCM_COSQ_GPORT_SCHEDULER_WFQ`	For scheduler resources
`BCM_COSQ_GPORT_ADD_COS`	
`BCM_COSQ_GPORT_CONF_SCOPE_CORE0`	
`BCM_COSQ_GPORT_CONF_SCOPE_CORE1`	

A new flag for `bcm_cosq_delay_tolerance_t` has been added.

Table 14: flags supported by the flags field of bcm_cosq_delay_tolerance_t

Flag	***Description***
`BCM_COSQ_DELAY_TOLERANCE_IS_LOW_LATENCY`	Specifies if the delay tolerance is low latency (multiple packet dequeue is enabled).

A new threshold type bcmCosqThresholdDynamicWeight has been added in this release.

Threshold Valid Bitmaps have been defined along with the three new fields of `bcm_cosq_threshold_t` and COSQ threshold initialization function.

Table 15: Threshold Valid Bitmamp

BCM_COSQ_THRESHOLD_VALID_DP	***dp field should be be used***
`BCM_COSQ_THRESHOLD_VALID_VALUE`	value field should be be used
`BCM_COSQ_THRESHOLD_VALID_TC`	tc field should be be used
`BCM_COSQ_THRESHOLD_VALID_VSQ_CATEGORY`	vsq category field should be be used

```
/* flow control / admission settings */
typedef struct bcm_cosq_threshold_s {
    uint32 flags;                   /* BCM_COSQ_THRESHOLD_* flag values */
    bcm_color_t dp;                 /* drop precedence. relevant for
                                 * BCM_COSQ_THRESHOLD_PER_DP flag value
                                     */
    bcm_cosq_threshold_type_t type; /* threshold type */
    int value;                     /* threshold value for drop/flow control */
    int tc;                       /* traffic class for which the thresholds are
                                 * to be configured
                                     */
    int vsq_category;              /* VSQ category for which the thresholds are
                                 * to be configured
                                     */
    uint32 valid;                   /* a bitmap representing enabled fields in
                                   the struct. */
} bcm_cosq_threshold_t;
```

bcm_cosq_threshold_t_init

Initialize a COSQ threshold structure.

Syntax

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

Parameters

`profile`	Pointer to threshold structure to initialize.

Description

Initialize a COSQ threshold structure.

Returns

None.

In addition, some other new COSQ APIs have been added.

bcm_cosq_gport_handle_core_get

Getting a gport handle

Syntax

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

int bcm_cosq_gport_handle_core_get(
    int unit,
    int core,
    bcm_cosq_gport_info_core_t *gport_info,
    bcm_gport_t *out_gport);
```

Parameters

`unit`	(IN) Unit number.
`core`	(IN) core number. For all cores use BCM_CORE_ALL
`gport_info`	(IN) gport info.
`out_gport`	(OUT) out gport.

Description

Getting a gport handle per core. This could be either a global gport (single instance on a device) or converting a gport to the desired type

Table 16: bcm_cosq_gport_info_core_t information for getting handle per core

Parameter	*Description*
`gport_type`	the gport type.
`in_gport`	gport to be converted. `NULL` for global gports (that is single instance on a device)
cosq	Num of cosq levels
flags	additional flags

Returns

`BCM_E_XXX`

bcm_cosq_bst_profile_t_init

Initialize a CoSQ BST profile structure.

Syntax

```
#include <bcm/cosq.h>
void bcm_cosq_bst_profile_t_init(bcm_cosq_bst_profile_t *profile);
```

Parameters

`profile`	Pointer to BST profile configuration structure to initialize.

Description

Initialize a CoSQ BST profile structure.

Returns

None.

bcm_cosq_bst_stat_get32

Get the current statistic/count of specified BST profile

Syntax

```
        #include <bcm/cosq.h>
      int bcm_cosq_bst_stat_get32(int unit, bcm_gport_t gport, bcm_cos_queue_t
cosq,
                        bcm_bst_stat_id_t bid, uint32 options, uint32 *pvalue);
```

Parameters

`unit`	BCM device number
`port`	Device or logical port or GPORT ID
`cosq`	Cosq object offset identifier
`bid`	BST stat ID to identify the COSQ resource/object
`options`	options to perform clear-on-read
`pvalue`	stat value to return.

Description

Retrieve the BST (buffer statistics tracking) statistic for the specified MMU resource. MMU resource is identified by the combination of port, cosq and bid parameters. port parameter can be port gport, queue gport. bid parameter identifies the MMU resource, for instance the flag bcmBstStatIdEgrPool identifies the egress service pool resource. cosq parameter identifies the object withing the various instances of resources within the identified resource. if option `BCM_COSQ_STAT_CLEAR` is present, a memory/register clear will be performed after stat reading.

Returns

`BCM_E_NONE`

`BCM_E_XXX`

bcm_cosq_bst_stat_multi_get32

Get the current statistic/count of multiple specified BST profile

Syntax

```
        #include <bcm/cosq.h>
        int bcm_cosq_bst_stat_multi_get32(int unit, bcm_gport_t gport,
bcm_cos_queue_t cosq,
                            uint32 options, int max_values, bcm_bst_stat_id_t
*id_list, uint32 *values);
```

Parameters

`unit`	BCM device number
`gport`	Device or logical port or GPORT ID
`cosq`	Cosq object offset identifier
`options`	options to perform clear-on-read
`max_values`	Number of elements in id_list and pvalue
`id_list`	Array of BST stat ID list to identify the COSQ resource/object
`pvalue`	Array of stat value to return.

Description

Retrieve the BST (buffer statistics tracking) statistic for multiple specified MMU resource. MMU resource is identified by the combination of port, cosq and bid parameters. port parameter can be port gport, queue gport. bid parameter identifies the MMU resource, for instance the flag bcmBstStatIdEgrPool identifies the egress service pool resource. cosq parameter identifies the object withing the various instances of resources within the identified resource. if option `BCM_COSQ_STAT_CLEAR` is present, a memory/register clear will be performed after stat reading.

Returns

`BCM_E_NONE`

`BCM_E_XXX`

bcm_cosq_service_pool_set
bcm_cosq_service_pool_get

To set/get enable or disable status of service pool for a given pool id

Syntax

```
        #include <bcm/cosq.h>
        int bcm_cosq_service_pool_set(
            int unit,
            bcm_service_pool_id_t id,
            bcm_cosq_service_pool_t cosq_service_pool);
        int bcm_cosq_service_pool_get(
            int unit,
            bcm_service_pool_id_t id,
            bcm_cosq_service_pool_t *cosq_service_pool);
```

Parameters

`unit`	(IN) BCM device number
`id`	(IN) Service pool id
`cosq_service_pool`	(IN) (for "_set") Enable/Disable given type service pool
`cosq_service_pool`	(OUT) (for "_get") Get enable or disable status on given

Description

To set/get enable or disable status on service pool on the specified pool id.

```
typedef struct bcm_cosq_service_pool_s {
    bcm_cosq_service_pool_type_t type;  /* Service pool type. */
    int enabled;                        /* Enable or disable. */
} bcm_cosq_service_pool_t;
```

Table 17: bcm_cosq_service_pool_type_t Service pool types

Parameter	***Description***
bcmCosqServicePoolPortColorAware	Port service pool level color aware
bcmCosqServicePoolColorAware	Pool level color aware
bcmCosqServicePoolQueueCongestionAware	Pool level queue congeston status aware

Returns

`BCM_E_NONE`

`BCM_E_XXX`

bcm_cosq_gport_reattach

Reattach the output of a GPORT to the input of a scheduler GPORT.

Syntax

```
        #include <bcm/cosq.h>
        int bcm_cosq_gport_reattach(int unit, bcm_gport_t sched_port,
                          bcm_gport_t input_port, bcm_cos_queue_t cosq);
```

Parameters

`unit`	(IN) BCM device number
`sched_port`	(IN) Scheduler GPORT ID
`input_port`	(IN) GPORT ID that will attach to scheduler gport
`cosq`	(IN) CoS queue to attach to

Description

This API can be used to reattach the output of a GPORT to the input of a scheduler GPORT to form a queuing hierarchy. The connect is done at the specified CoS queue of the Scheduler GPORT. A `cosq` value of -1 will attach the input GPORT to the first unused CoS queue of the Scheduler GPORT (going from higher to lower CoS queue priority).

Returns

```
BCM_E_NONE
BCM_E_XXX
```

bcm_cosq_profile_mapping_set

Maps a gport to a profile

Syntax

```
#include <bcm/cosq.h>
int 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);
int bcm_cosq_profile_mapping_get(
    int unit,
    bcm_gport_t gport_to_map,
    bcm_cos_queue_t cosq,
    uint32 flags,
    bcm_switch_profile_mapping_t* profile_mapping);
```

Parameters

`unit`	(IN) BCM device number
`gport_to_map`	(IN) Gport to map
`cosq`	(IN) CoS queue to attach to
`flags`	(IN) Flags
`profile_mapping`	(IN) (for "_set") Map gport into profile
`profile_mapping`	(OUT) (for "_get") The profile that the gport is mapped into

Description

Mapping of a gport to a profile. Where `gport_to_map` is a handle of some object in the CosQ pipe that is being profiled be an entity defined by `profile_mapping->mapped_profile`.

```
/* profile mapping type. */
typedef enum bcm_switch_profile_mapping_type_e {
    bcmbcmCosqIngressQueueToRateClass = 0 /* ingress queue to rate class
                                           * mapping
                                           */
} bcm_switch_profile_mapping_type_t;

/* profile mapping struct. */
typedef struct bcm_switch_profile_mapping_s {
    /* The type of profile to be mapped to. */
    bcm_switch_profile_mapping_type_t profile_type;
    bcm_gport_t mapped_profile;            /* mapped profile handle */
} bcm_switch_profile_mapping_t;
```

Returns

`BCM_E_NONE`

`BCM_E_XXX`

Two dynamic scheduling APIs with dynamic weights types have been added in this release

bcm_cosq_gport_dynamic_sched_set
bcm_cosq_gport_dynamic_sched_get

set/get weights dynamic or regular weights

Syntax

```
#include <bcm/cosq.h>
bcm_cosq_gport_dynamic_sched_set(int unit, bcm_gport_t gport, bcm_cos_queue_t
cosq,
 bcm_cosq_dynamic_state_t state, int weight);

#include <bcm/cosq.h>
bcm_cosq_gport_dynamic_sched_get(int unit, bcm_gport_t gport, bcm_cos_queue_t
cosq,
 bcm_cosq_dynamic_state_t state, int *weight);
```

Parameters

`unit`	BCM device number
`gport`	gport type, see table
`cosq`	cosq num
`state`	type of weight to set set according to congestion state
`weigth`	corresponding weight.

Description

get/set dynamic weights. The type of the weight is determined by "state"

Table 18: dynamic weights types

bcmCosqDynamicStateNormal	***regular state***
bcmCosqDynamicStateQueue0Congested	pipe 0 is congested, pipe 1 not congested
bcmCosqDynamicStateQueue1Congested	pipe 1 is congested, pipe 0 not congested

bcm_cosq_gport_stat_sync_get

Force an immediate counter update and retrieve statistics.

Syntax

```
#include <bcm/cosq.h>
int
bcm_cosq_gport_stat_sync_get(int unit, bcm_gport_t gport,
     bcm_cos_queue_t cosq,
     bcm_cosq_gport_stats_t stat,
     uint64 *value);
```

Parameters

`unit`	(IN) Unit number.
`gport`	(IN) GPORT ID for a queue group
`cosq`	(IN) CoS Queue
`stat`	(IN) statistic to be retrieved.
`value`	(OUT) (for "_get") statistic value

Description

Similar to `bcm_cosq_gport_stat_get()`, value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_XXX
```

PORT EXTENSION MANAGEMENT

A new `qos_map_id` filed of `bcm_extender_port_t` has been introduced for the egress etag mapping. The `bcm_extender_port_t` structure describes a Port Extender port logically attached to a Controlling Bridge.

```
typedef struct bcm_extender_port_s {
    ...
    int qos_map_id;    /* Qos map id for egress etag mapping profile. */
    ...
} bcm_extender_port_t;
```

FABRIC

A new fabric control type bcmFabricForceTdmBypassTrafficToFabric has been added for `bcm_fabric_control_get/set()` functions.

Table 19: Fabric Type Values

Value	Description	Arg Value
bcmFabricForceTdmBypassTrafficToFabric	Enable/Disable Forcing of TDM bypass traffic to fabric	1/0

A new fabric link control type has been added for `bcm_fabric_link_control_set()` and `bcm_fabric_link_control_get()` APIs.

Table 20: bcm_fabric_link_control_t

Control Type	Description	Arg Expected Values
bcmFabricLinkPcpEnable	Enable/Disable Packet Cell Packing	1: the remote device support PCP 0: otherwise

Several fabric link threshold control types have been added for `bcm_fabric_link_thresholds_get/set` functions.

Table 21: bcm_fabric_link_threshold_type_t

enum	Description
bcmFabricLinkRxRciLvl1FC	The RX FIFO threshold that initiates the Route Congestion Indication level 1 flow control
bcmFabricLinkRxRciLvl2FC	The RX FIFO threshold that initiates the Route Congestion Indication level 2 flow control
bcmFabricLinkRxRciLvl3FC	The RX FIFO threshold that initiates the Route Congestion Indication level 3 flow control
bcmFabricLinkRxFull	Configure threshold for max fifo size, beyond this threshold, packets will be dropped.

Table 21: bcm_fabric_link_threshold_type_t

enum	Description
bcmFabricLinkRxFifoSize	Configure the FIFO size, this is a static configuration.
bcmFabricLinkRxMcLowPrioDrop	Configure the multicast low priority cells drop thresholds.
bcmFabricLinkMidGciLvl1FC	Configure the Mid GCI threshold for level 1 flow control.
bcmFabricLinkMidGciLvl2FC	Configure the Mid GCI threshold for level 2 flow control.
bcmFabricLinkMidGciLvl3FC	Configure the Mid GCI threshold for level 3 flow control.
bcmFabricLinkMidRciLvl1FC	Configure the Mid RCI threshold for level 1 flow control.
bcmFabricLinkMidRciLvl2FC	Configure the Mid RCI threshold for level 2 flow control.
bcmFabricLinkMidRciLvl3FC	Configure the Mid RCI threshold for level 3 flow control.
bcmFabricLinkMidPrio0Drop	Configure the maximum threshold for DCM priorty 0 drops, above this threshold the dcm will drop cells recieved from DCH.
bcmFabricLinkMidPrio1Drop	Configure the maximum threshold for DCM priorty 1 drops, above this threshold the dcm will drop cells recieved from DCH.
bcmFabricLinkMidPrio2Drop	Configure the maximum threshold for DCM priorty 2 drops, above this threshold the dcm will drop cells recieved from DCH.
bcmFabricLinkMidPrio3Drop	Configure the maximum threshold for DCM priorty 3 drops, above this threshold the dcm will drop cells recieved from DCH.
bcmFabricLinkMidAlmostFull	Configure threshold for almost full, beyond this threshold, flow control will be sent back to the DCH.
bcmFabricLinkMidFifoSize	Configure the FIFO depth (size).
bcmFabricLinkTxRciLvl1FC	The TX FIFO threshold that initiates the Route Congestion Indication level 1 Flow Control
bcmFabricLinkTxRciLvl2FC	The TX FIFO threshold that initiates the Route Congestion Indication level 2 Flow Control
bcmFabricLinkTxRciLvl3FC	The TX FIFO threshold that initiates the Route Congestion Indication level 3 Flow Control
bcmFabricLinkTxAlmostFull	Configure threshold for almost full, beyond this threshold, flow control will be sent back to the DCM.
bcmFabricLinkTxFifoSize	Configure DCL FIFO depth (size).

The new functions to get/set the `fifo_type` thresholds per pipe have been added as well as three fabric route functions have been added with required data structure.

bcm_fabric_link_thresholds_pipe_set bcm_fabric_link_thresholds_pipe_get

SetGet fifo_type thresholds per pipe

Syntax

```
#include <bcm/fabric.h>
int
bcm_fabric_link_thresholds_pipe_set(
    int unit,
    int fifo_type,
    bcm_fabric_pipe_t pipe,
    uint32 flags,
    uint32 count,
    bcm_fabric_link_threshold_type_t *type,
    int *value)

int
bcm_fabric_link_thresholds_pipe_get(
    int unit,
    int fifo_type,
    bcm_fabric_pipe_t pipe,
    uint32 flags,
    uint32 count,
    bcm_fabric_link_threshold_type_t *type,
    int *value)
```

Parameters

`unit`	(IN) Unit number.
`fifo_type`	(IN) fifo_type you would like to update (0 or 1)
`pipe`	(IN) which pipe you would like to update
`flags`	(IN) relevant flags: (BCM_FABRIC_LINK_TH_FE1_LINKS_ONLY/ BCM_FABRIC_LINK_TH_FE3_LINKS_ONLY/none)
`count`	(IN) Size of the array
`type`	(IN) Array of bcm_fabric_link_threshold_type_t to get or set
`value`	(IN) (for "_set") value[i] is the threshold to be assigned with type[i]
`value`	(OUT) (for "_get") value[i] is the threshold assigned with type[i]

Description

For `fifo_type` 0 or 1 , set the relevant set of thresholds. See `bcm_fabric_link_threshold_type_t` (page 26) br Use a relevant pipe with: br

Table 22: bcm_fabric_pipe_t

enum	*Description*
bcmFabricPipe0	Pipe 0
bcmFabricPipe1	Pipe 1
bcmFabricPipe2	Pipe 2
bcmFabricPipeAll	All pipes

Returns

`BCM_E_xxx`

bcm_fabric_route_t_init

Initialize a bcm_fabric_route_t to a 'safe' default value.

Syntax

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

        void
        bcm_fabric_route_t_init(bcm_fabric_route_t *fabric_route)
```

Parameters

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

Description

Sets a 'safe' default state for a `bcm_fabric_route_t.`

Returns

None.

bcm_fabric_route_tx

Send fabric route cells

Syntax

```
        #include <bcm/fabric.h>
        int
        bcm_fabric_route_tx(int unit,
                            uint32 flags,
                            bcm_fabric_route_t *route,
                            uint32 data_in_size,
                            uint32 *data_in)
```

Parameters

`unit`	(IN) Unit number.
`flags`	(IN) flags
`route`	(IN) Specify the path for the route cells.
`data_in_size`	(IN) "data_in" size (uint32 units)
`data_in`	(IN) The data to send.

Description

Fabric route cells, generated by CPU, can be transmitted and received (over fabric interface). These cells are routed according to the specific path information they carry. These messages are used mainly for debug and diagnostics purposes, but can be also used for CPU-to-CPU messaging.

Returns

`BCM_E_XXX`

bcm_fabric_route_rx

Receive fabric route cells

Syntax

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

        int
        bcm_fabric_route_rx(int unit,
                            uint32 flags,
                            uint32 data_out_max_size,
                            uint32 *data_out,
                            uint32 *data_out_size)
```

Parameters

unit	(IN) Unit number
flags	(IN) flags
data_out_max_size	(IN) max "data_out" size (uint32 units)
data_out	(OUT) the received data
data_out_size	(OUT) "data_out" actual size (uint32 units)

Description

Fabric route cells, generated by CPU, can be transmitted and received (over fabric interface). These cells are routed according to the specific path information they carry. These messages are used mainly for debug and diagnostics purposes, but can be also used for CPU-to-CPU messaging.

Returns

```
BCM_E_XXX
```

Fabric Pipe Mapping has been introduced with the required data types in this release.

bcm_fabric_link_remote_pipe_mapping_set
bcm_fabric_link_remote_pipe_mapping_get

Set/get per link the mapping between the local pipe to the remote pipe.

Syntax

```
#include <bcm/fabric.h>
int bcm_fabric_link_remote_pipe_mapping_set(
    int unit,
    bcm_port_t port,
    bcm_fabric_link_remote_pipe_mapping_t *mapping_config);

int bcm_fabric_link_remote_pipe_mapping_get(
    int unit,
    bcm_port_t port,
    bcm_fabric_link_remote_pipe_mapping_t *mapping_config);

typedef struct bcm_fabric_link_remote_pipe_mapping_s {
    uint32 num_of_remote_pipes;     /* Number of pipes supported by the
    * remote device */
    bcm_fabric_pipe_t remote_pipe_mapping;
                                    /* The mapping from remote pipe
    * (represented by the index) to the
    * local pipe (represented by the value
    * stored in this index)
                                     */
    uint32 remote_pipe_mapping_max_size;
                                    /* Max size of remote_pipe_mapping */
} bcm_fabric_link_remote_pipe_mapping_t;
```

Parameters

`unit`	(IN) Unit number.
`port`	(IN) Port number.
`mapping_config`	(IN) Specify eterive the mapping between the remote pipe to local pipe

Description

Set/get per link the mapping between the local pipe to the remote pipe. Should be used in a case the remote pipe configuration is a sub set of local pipe configuration.

Returns

```
BCM_E_XXX
```

bcm_fabric_link_remote_pipe_mapping_t_init

Initialize a bcm_fabric_link_remote_pipe_mapping_t to a 'safe' default value.

Syntax

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

void
bcm_fabric_link_remote_pipe_mapping_t_init(bcm_fabric_link_remote_pipe_mapping
_t *pipe_mapping)
```

Parameters

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

Description

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

Returns

void

New APIs of PCP mode config get and set functions for PCP mode per VOQ/destination device are added with the data types.

bcm_fabric_pcp_dest_mode_config_set bcm_fabric_pcp_dest_mode_config_get

Set/Get PCP mode per VOQ/destination device.

Syntax

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

int
bcm_fabric_pcp_dest_mode_config_set(
    int unit,
    uint32 flags,
    bcm_module_t modid,
    bcm_fabric_pcp_mode_config_t *pcp_config)

int
bcm_fabric_pcp_dest_mode_config_get(
    int unit,
    uint32 flags,
    bcm_module_t modid,
    bcm_fabric_pcp_mode_config_t *pcp_config)

use this struct in order to configure which fabric_pcp mode
is used for a certain destination device.


/* Fabric pcp mode configuration */
typedef struct bcm_fabric_pcp_mode_config_s {
    bcm_fabric_pcp_mode_t pcp_mode; /* Mode of Packet Cell Packing (PCP) */
} bcm_fabric_pcp_mode_config_t;
```

Parameters

`unit`	(IN) Unit number
`flags`	(IN) Additional flags
`modid`	(IN) Id of destination device
`pcp_config`	(IN/OUT) Pointer to configuration struct which the fabric-PCP mode is set from/ reterived to.

Description

Set/Get the Packet Cell Packing mode for a specific destination device.

Returns

```
BCM_E_XXX
```

FORWARDING FAILOVER PROTECTION

Below new Failover Create flags are added in this release.

Table 23: Failover Create Flags

FLAG	*Meaning*
`BCM_FAILOVER_PAIRED`	flag to allocate a pair of adjoining failover objects
`BCM_FAILOVER_FEC`	flag to create a FEC failover object for 1:1 UC Protection
`BCM_FAILOVER_INGRESS`	flag to create an Ingress failover object for 1+1 Protection
`BCM_FAILOVER_ENCAP`	flag to create an encapsulation failover object for 1:1 MC Protection

FCMAP API

New APIs `bcm_fcmap_linkfault_trigger_rc_get()` and `bcm_fcmap_diag_get()` have been added with the associated data structures.

bcm_fcmap_linkfault_trigger_rc_get

Returns the linkfault trigger code and reason code from the specified FC port.

Syntax

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

 int bcm_fcmap_linkfault_trigger_rc_get(int unit,
bcm_port_t port,
bcm_fcmap_lf_tr_t *trigger,
bcm_fcmap_lf_rc_t *rc);
```

Parameters

`unit`	(IN) BCM unit
`port`	(IN) Port identifier
`trigger`	(OUT) Pointer to the to be returned linfault trigger code
`rc`	(OUT) Pointer to the to be returned linkfault reason code

Description

This API returns the linkfault trigger code and reason code from the specified FC port.

```
/* link fault trigger. */
#define BCM_FCMAP_LF_TR_NONE                           0x0
#define BCM_FCMAP_LF_TR_PORT_INIT                      0x1
#define BCM_FCMAP_LF_TR_OPEN_LINK                      0x2
```

```
#define BCM_FCMAP_LF_TR_LINK_FAILURE                0x3
#define BCM_FCMAP_LF_TR_OLS_RCVD                    0x4
#define BCM_FCMAP_LF_TR_NOS_RCVD                    0x5
#define BCM_FCMAP_LF_TR_SYNC_LOSS                   0x6
#define BCM_FCMAP_LF_TR_BOUCELINK_FROM_ADMIN        0x7
#define BCM_FCMAP_LF_TR_CHGSPEED_FROM_ADMIN         0x8
#define BCM_FCMAP_LF_TR_DISABLE_FROM_ADMIN          0x9
#define BCM_FCMAP_LF_TR_RESET_FROM_ADMIN            0xA
#define BCM_FCMAP_LF_TR_LR_RCVD                     0xB
#define BCM_FCMAP_LF_TR_LRR_RCVD                    0xC
#define BCM_FCMAP_LF_TR_ED_TOV                      0xD

/* link fault reason code. */
#define BCM_FCMAP_LF_RC_NONE                       0x0
#define BCM_FCMAP_LF_RC_PORT_INIT                  0x1
#define BCM_FCMAP_LF_RC_OPEN_LINK                  0x2
#define BCM_FCMAP_LF_RC_LINK_FAILURE               0x3
#define BCM_FCMAP_LF_RC_OLS_RCVD                   0x4
#define BCM_FCMAP_LF_RC_NOS_RCVD                   0x5
#define BCM_FCMAP_LF_RC_SYNC_LOSS                  0x6
#define BCM_FCMAP_LF_RC_BOUCELINK_FROM_ADMIN       0x7
#define BCM_FCMAP_LF_RC_CHGSPEED_FROM_ADMIN        0x8
#define BCM_FCMAP_LF_RC_DISABLE_FROM_ADMIN         0x9
#define BCM_FCMAP_LF_RC_RESET_FAILURE              0xA
```

Returns

`BCM_E_XXX`	bcm_error_t type

bcm_fcmap_diag_get

Returns the current diagnostic code from the specified FC port.

Syntax

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

       int bcm_fcmap_diag_get(int unit, bcm_port_t port,
      bcm_fcmap_diag_code_t *diag);
```

Parameters

`unit`	(IN) BCM unit
`port`	(IN) Port identifier
`diag`	(OUT) Pointer to the to be returned diagnostic code

Description

This API returns the current diagnostic code from the specified FC port. The diagnostic code is returned in the user-supplied diag variable.

```
#define BCM_FCMAP_DIAG_OK                                        0x0
#define BCM_FCMAP_PORT_INIT                                      0x1
#define BCM_FCMAP_OPEN_LINK                                      0x2
#define BCM_FCMAP_LINK_FAILURE                                   0x3
#define BCM_FCMAP_OLS_RCVD                                       0x4
#define BCM_FCMAP_NOS_RCVD                                       0x5
#define BCM_FCMAP_SYNC_LOSS                                      0x6
#define BCM_FCMAP_BOUCELINK_FROM_ADMIN                           0x7
#define BCM_FCMAP_CHGSPEED_FROM_ADMIN                            0x8
#define BCM_FCMAP_DISABLE_FROM_ADMIN                             0x9
#define BCM_FCMAP_AN_NO_SIGNAL                                   0xA
#define BCM_FCMAP_AN_TIMEOUT                                     0xB
#define BCM_FCMAP_PROTO_TIMEOUT                                  0xC
```

Returns

`BCM_E_XXX`	bcm_error_t type

FIBER CHANNEL OVER ETHERNET

New APIs `bcm_fcoe_vsan_stat_counter_sync_get()` and `bcm_fcoe_route_stat_counter_sync_get()` have been added.

bcm_fcoe_vsan_stat_counter_sync_get

Force an immediate counter update and retrieve flex counter value for FCOE VSAN ID.

Syntax

```
#include <bcm/fcoe.h>
int
bcm_fcoe_vsan_stat_counter_sync_get(int                 unit,
    bcm_fcoe_vsan_id_t   vsan,
    bcm_fcoe_vsan_stat_t stat,
    uint32               num_entries,
    uint32               *counter_indexes,
    bcm_stat_value_t     *counter_values);
```

Parameters

`unit`	BCM device number
`vsan`	VSAN ID
`stat`	packet or byte count selector
`num_entries`	number of entries to get
`counter_indexes`	pointer to indexes
`counter_values`	pointer to return value structures

Description

Similar to `bcm_fcoe_vsan_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_XXX
```

bcm_fcoe_route_stat_counter_sync_get

Force an immediate counter update and retrieve flex counter
value for FCOE route.

Syntax

```
#include <bcm/fcoe.h>
int
bcm_fcoe_route_stat_counter_sync_get(int                   unit,
     bcm_fcoe_route_t      *route,
     bcm_fcoe_route_stat_t stat,
     uint32                num_entries,
     uint32                *counter_indexes,
     bcm_stat_value_t      *counter_values);
```

Parameters

`unit`	BCM device number
`route`	FCOE route
`stat`	packet or byte count selector
`num_entries`	number of entries to get
`counter_indexes`	pointer to indexes
`counter_values`	pointer to return value structures

Description

Similar to `bcm_fcoe_route_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_XXX
```

FIELD PROCESSOR

Several flags regarding packet content (data) qualification have been added along with the new field flags in `bcm_field_data_packet_format_t.`

Table 24: New flags of packet content (data) qualification

Misc Flags	*Purpose*
BCM_FIELD_DATA_FORMAT_F_NO_ENCAP_STACK	Indicates incoming packet is not a Higig Packet.
BCM_FIELD_DATA_FORMAT_F_VNTAG	Indicates incoming packet is a Vntag Packet.
BCM_FIELD_DATA_FORMAT_F_NO_VNTAG	Indicates incoming packet is not a Vntag Packet.
BCM_FIELD_DATA_FORMAT_F_CNTAG	Indicates incoming packet is a Ctag Packet.
BCM_FIELD_DATA_FORMAT_F_NO_CNTAG	Indicates incoming packet is not a Ctag Packet.
BCM_FIELD_DATA_FORMAT_F_ETAG	Indicates incoming packet is a Etag Packet.
BCM_FIELD_DATA_FORMAT_F_NO_ETAG	Indicates incoming packet is not a Etag Packet.
BCM_FIELD_DATA_FORMAT_F_ICNM	Indicates incoming packet is a Icnm Packet.
BCM_FIELD_DATA_FORMAT_F_NO_ICNM	Indicates incoming packet is not a Icnm Packet.

```
/* Packet format based DATA qualifier specification structure. */
typedef struct bcm_field_data_packet_format_s {
   ...
   uint32 flags;            /* Flags. (FIELD_DATA_FORMAT_F_XXX) */
} bcm_field_data_packet_format_t;
```

New BCM Field Qualifier has been added while the flag bcmFieldQualifySrcGport of `bcm_field_qualify_t` is now applied for NiV port.

Table 25: Field Qualifiers

Qualifier	*Purpose*
bcmFieldQualifySrcNivGport	Source Niv gport.
bcmFieldQualifyDstNivGport	Destination Niv gport.
bcmFieldQualifyDstGport	Destination module/port gport or MPLS/MiM/WLAN/Niv gport.
bcmFieldQualifyMirrorEgressDisabled	Egress Mirroring disabled.
bcmFieldQualifyBypassSrcMacFilter	Qualify packets bypassing Source MAC Address filtering.
bcmFieldQualifyRxTrapCodeForSnoop	Rx Trap code ID used for snoop.
bcmFieldQualifyIpmcStarGroupValue	L3 Multicast lookup (Star,Group) Value.
bcmFieldQualifyNativeVSwitch	Native-VSwitch (for Routing-over-overlay packets).
bcmFieldQualifyNativeVSwitchValid	Valid native-VSwitch (for Routing-over-overlay packets).
bcmFieldQualifyTrillVersion	TRILL version field.

Table 25: Field Qualifiers

Qualifier	Purpose
bcmFieldQualifyTrillIngressRbridge	Ingress Rbridge Nickname.
bcmFieldQualifyTrillMultiDestination	TRILL Multi-destination field.
bcmFieldQualifyTrillOpLength	TRILL Options length field.
bcmFieldQualifyTrillHopCount	TRILL Hop count field.
bcmFieldQualifyTrillPayload	Payload of TRILL header.
bcmFieldQualifyMplsForwardingLabelPayload	Payload of MPLS forwarding label.
bcmFieldQualifyIncomingIpIfClass	Incoming IP interface Class ID.
bcmFieldQualifyL2GreProtocolType	L2GRE Protocol type field.
bcmFieldQualifyL2GreVsid	L2GRE Virtual Subnet ID field.
bcmFieldQualifyL2GreFlowId	L2GRE Flow ID field.
bcmFieldQualifyMimVlanPri	MAC-in-MAC VLAN Priority field.
bcmFieldQualifyMimVlanCfi	MAC-in-MAC VLAN Cfi field.
bcmFieldQualifyFhei	DNX FHEI header field.
bcmFieldQualifyFheiSize	DNX FHEI header size in bytes.
bcmFieldQualifyEthernetOamHeaderBits0_31	Ethernet OAM Header First 4 Bytes.
bcmFieldQualifyEthernetOamHeaderBits32_63	Ethernet OAM Header Second 4 Bytes.
bcmFieldQualifyEthernetOamDstClassL2	Ethernet OAM Destination Class based on L2 lookup result.
bcmFieldQualifyEthernetOamTxPktUPMEP	Ethernet OAM UP-MEP Tx Packet.
bcmFieldQualifyEthernetOamSrcPortLmStatPoolId	Ethernet OAM LM Stat PoolId based on SrcPort.
bcmFieldQualifyEthernetOamVxlt1LmStatPoolId	Ethernet OAM LM Stat PoolId based on 1st Lookup in Vlan_Xlate table.
bcmFieldQualifyEthernetOamVxlt2LmStatPoolId	Ethernet OAM LM Stat PoolId based on 2nd Lookup in Vlan_Xlate table
bcmFieldQualifyEthernetOamSourceVPLmStatPoolId	Ethernet OAM LM Stat PoolId based on Source VP table.
bcmFieldQualifyEthernetOamInterfaceClassPort	Ethernet OAM Class Id assigned for packet based on Ingress Port.
bcmFieldQualifyEthernetOamClassVxlt1	Ethernet OAM Class Id assigned for packet based on 1st Lookup in Vlan_Xlate table.
bcmFieldQualifyEthernetOamClassVxlt2	Ethernet OAM Class Id assigned for packet based on 2nd Lookup in Vlan_Xlate table.
bcmFieldQualifyEthernetOamClassSourceVP	Ethernet OAM Class Id assigned for packet based on Source VP table.
bcmFieldQualifyEthernetOamClassVFI	Ethernet OAM Class Id assigned for packet based on VFI table.
bcmFieldQualifyMplsOamHeaderBits0_31	Mpls OAM Header First 4 Bytes.
bcmFieldQualifyMplsOamHeaderBits32_63	Mpls OAM Header Second 4 Bytes.
bcmFieldQualifyMplsOamGALLabelOnly	MPLS OAM packets having GAL Label Only.
bcmFieldQualifyMplsOamLmStatPoolId	MPLS OAM LM Stat PoolId based on MPLS_ENTRY table.
bcmFieldQualifyMplsOamGALExposed	MPLS OAM packets having GAL exposed.
bcmFieldQualifyMplsOamACH MPLS OAM ACH Header.	
bcmFieldQualifyMplsOamControlPktType	MPLS OAM control packets based on Packet Type.
bcmFieldQualifyMplsOamClassMpls	MPLS OAM Class Id assigned for packet based on MPLS table.
bcmFieldQualifyOamHeaderBits0_31	OAM Header First 4 Bytes.
bcmFieldQualifyOamHeaderBits32_63	OAM Header Second 4 Bytes.
bcmFieldQualifyOamEgressClassVxlt	OAM Class Id assigned for packet based on EGR_VXLT table.
bcmFieldQualifyOamEgressClassPort	OAM Class Id assigned for packet based on Port table.

Table 25: Field Qualifiers

Qualifier	***Purpose***
bcmFieldQualifyUdf	Qualify UDF selected data
bcmFieldQualifyOuterVlanActionRange	Compressed outer vlan tag
bcmFieldQualifyInnerVlanActionRange	Compressed inner vlan tag

The new field actions have been added for `bcm_field_action_t`

Table 26: Field Actions

Action	***Description***	***param0***	***param1***
bcmFieldActionRedirDropPrecedence	Set the redirect drop precedence.	`BCM_FIELD_COLOR_xxx`	n/a
bcmFieldActionRpRedirDropPrecedence	Set the redirect drop precedence for Red packets.	`BCM_FIELD_COLOR_xxx`	n/a
bcmFieldActionYpRedirDropPrecedence	Set the redirect drop precedence for Yellow packets.	`BCM_FIELD_COLOR_xxx`	n/a
bcmFieldActionGpRedirDropPrecedence	Set the redirect drop precedence for Green packets.	`BCM_FIELD_COLOR_xxx`	n/a
bcmFieldActionHiGigDstModuleGportNew	Replace the value of `DST_MODID` field in Higig header.	New Destination Module Id.	n/a
bcmFieldActionHiGigDstPortGportNew	Replace the value of `DST_PORT` field in Higig header.	New Destination Port Id.	n/a
bcmFieldActionHiGigDstGportNew	Replace the value of `DST_MODID` and `DST_PORT` fields in Higig header	New Destination Gport.	n/a
bcmFieldActionGpHiGigDropPrecedenceNew	Replace the value of Drop Precedence field in Higig header for Green Packets.	`BCM_FIELD_COLOR_xxx.`	n/a
bcmFieldActionYpHiGigDropPrecedenceNew	Replace the value of Drop Precedence field in Higig header for Yellow packets	`BCM_FIELD_COLOR_xxx.`	n/a
bcmFieldActionRpHiGigDropPrecedenceNew	Replace the value of Drop Precedence field in Higig header for Red packets	param0: `BCM_FIELD_COLOR_xxx.`	n/a
bcmFieldActionHiGigDropPrecedenceNew	Replace the value of Drop Precedence field in Higig header for any(Green/ Yellow/Red) packets	param0: `BCM_FIELD_COLOR_xxx.`	n/a
bcmFieldActionGpHiGigIntPriNew	Replace the value of Traffic Class field in Higig header for Green packets		
param0: New Internal Priority.	n/a		
bcmFieldActionYpHiGigIntPriNew	Replace the value of Traffic Class field in Higig header for Yellow packets	param0: New Internal Priority.	n/a
bcmFieldActionRpHiGigIntPriNew	Replace the value of Traffic Class field in Higig header for Red packets	param0: New Internal Priority.	n/a

Table 26: Field Actions

Action	Description	param0	param1
bcmFieldActionHiGigIntPriNew	Replace the value of Traffic Class field in Higig header for any(green/Yellow/Red) packet	param0: New Internal Priority.	n/a
bcmFieldActionLoopbackSrcModuleGportNew	Replace the value of Sourece ModId field in Loopback header	param0: New Source Module Id.	n/a
bcmFieldActionLoopbackSrcPortGportNew	Replace the value of Source PortId field in Loopback header	param0: New Source Port Id.	n/a
bcmFieldActionLoopbackSrcGportNew	Replace the value of Source ModId and Source PortId fields in Loopback header	param0: New Source Gport.	n/a
bcmFieldActionLoopbackCpuMasqueradePktProfileNew	Replace the value of Packet Profile field in Loopback header	param0: New Packet Profile.	n/a
bcmFieldActionLoopbackPacketProcessingPortNew	Replace the value of Packet Processing port field in Loopback header	param0: New Packet Processing Port.	n/a
bcmFieldActionLoopBackTypeNew	Replace the value of Loopback Type field in Loopback header	param0: Loopback Type (bcmFieldLoopbackTypeXXX).	n/a
bcmFieldActionRecoverableDropCancel	Override drop decision taken by Protection switching logic		

A new field control type has been added in `bcm_field_control_t`.

Table 27: Field Control Values

Name	Purpose
bcmFieldControlIngressLogicalPolicerPoolsMode	Configure Logical Policer pool mode.

FIELD QUALIFY VLANTRANSLATIONHIT

Matching on vlan translation hit status is done with the qualifier bcmFieldQualifyVlanTranslationHit.The possible values for the parameters of `bcm_field_qualify_VlanTranslationHit()` are defined as below.

Table 28: Vlan Translation Hit Status Values (for bcm_field_qualify_VlanTranslationHit)

BCM_FIELD_VXLT_LOOKUP_STATUS_xxx	Purpose
BCM_FIELD_VXLT_LOOKUP_STATUS_NO_HIT	No Hit on First/Second lookup of Vlan Translation Table
BCM_FIELD_VXLT_LOOKUP_STATUS_HIT	Hit on First/Second lookup of Vlan Translation Table
BCM_FIELD_VXLT_LOOKUP_STATUS_FIRST_HIT	Hit on First lookup of Vlan Translation Table
BCM_FIELD_VXLT_LOOKUP_STATUS_SECOND_HIT	Hit on Second lookup of Vlan Translation Table

One of the above mentioned constants has to be passed as data argument to `bcm_field_qualify_VlanTranslationHit()` API. But mask argument passed to the API will be ignored as `BCM_FIELD_VXLT_LOOKUP_STATUS_xxx` constants covers all possible combinations.

`bcm_field_qualify_VlanTranslationHit_get()` API retrieves actual data value configured by the set API, mask has no significance and is populated with all 1's.

INGRESS POLICER POOL MODES

Triumph3 has 16 Slices in Ingress Field Processor, with each slice having 512 entries. So the total number of entries comes to 8K. It has same number of meters (8K) which are spread across 8 global pools, with each pool having 1K meters (512 meter pairs). Triumph3 allows only 8 TCAM slices to update meters in the 8 meter pools for a matching packet. SDK assigns one meter pool for one group (i.e. one slice) and also restricts assigning meter pools to only 8 groups (8 slices). When ACL is configured in ways where some groups are mutually exclusive, those groups can share the same meter pool since a packet will be matching in only one of those groups. In such cases bcmFieldControlIngressLogicalPolicerPoolsMode can be used to divide each physical meter pools into two logical meter pools.

Table 29: Ingress Policer Pool Modes

Mode	***Purpose***
bcmFieldIngressLogicalPolicerPools8x1024	8 Meter pools with 1024 entries in each pool (default).
bcmFieldIngressLogicalPolicerPools16x512Contiguous	16 Meter pools with 512 entries in each pool in Contiguous mode.
bcmFieldIngressLogicalPolicerPools16x512Split	16 Meter pools with 512 entries in each pool in Split mode.
bcmFieldIngressPolicerPoolsCount	Internal use only

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

bcm_field_qset_id_multi_set

Add qualifier objects to field group qset.

Syntax

```
#include <bcm/field.h>
int
bcm_field_qset_id_multi_set(
    int unit,
    bcm_field_qualify_t qualifier,
    int num_objects,
    int *object_list,
    bcm_field_qset_t *qset);
```

Parameters

`unit`	(IN) Unit number.
`qualifier`	(IN) Qualifier enum bcmFieldQualifyXXX
`num_objects`	(IN) Number of objects to set in the qset
`object_list`	(IN) List of objects to set in the qset
`qset`	(IN/OUT) Field qualifier set - Qset

Description

Add qualifier objects to field group qset. The APIs `bcm_field_qset_data_qualifier_add()` and `bcm_field_qset_id_multi_set (`page 41`)` should not be intermixed.

Returns

`BCM_E_xxx`

bcm_field_qset_id_multi_delete

Delete qualifier objects to field group qset.

Syntax

```
#include <bcm/field.h>
int
bcm_field_qset_id_multi_delete(
    int unit,
    bcm_field_qualify_t qualifier,
    int num_objects,
    int *object_list,
    bcm_field_qset_t *qset);
```

Parameters

`unit`	(IN) Unit number.
`qualifier`	(IN) Qualifier enum bcmFieldQualifyXXX
`num_objects`	(IN) Number of objects to be removed from the qset
`object_list`	(IN) List of objects to be removed from the qset
`qset`	(IN/OUT) Field qualifier set - Qset

Description

Delete qualifier objects to field group qset.

Returns

`BCM_E_xxx`

bcm_field_qset_id_multi_get

Get qualifier objects added to the field group qset.

Syntax

```
#include <bcm/field.h>
int
bcm_field_qset_id_multi_get(
    int unit,
    bcm_field_qset_t qset,
    bcm_field_qualify_t qualifier,
    int max_objects,
    int *object_list,
    int *count_objects);
```

Parameters

`unit`	(IN) Unit number.
`qset`	(IN) Field qualifier set - Qset
`qualifier`	(IN) Qualifier enum bcmFieldQualifyXXX
`max_objects`	(IN) Number of objects to be fetched from the qset
`object_list`	(OUT) List of objects be fetched from the qset
`count_objects(OUT)`	Number of objects actually fetched from the qset

Description

Get qualifier objects added to the field group qset.

Returns

```
BCM_E_xxx
```

bcm_field_qualify_XXX

Add a qualification to a field entry

Syntax

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

        int bcm_field_qualify_IsEqualValue(
                int unit,
                bcm_field_entry_t entry,
                uint8 data,
                uint8 mask);
        int bcm_field_qualify_MirrorEgressDisabled(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 mask);
        int bcm_field_qualify_BypassSrcMacFilter(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 mask);
        int bcm_field_qualify_RxTrapCodeForSnoop(
        int unit,
        bcm_field_entry_t entry,
        uint32 data,
        uint32 mask);
        int bcm_field_qualify_IpmcStarGroupValue(
                int unit,
                bcm_field_entry_t entry,
                uint64 data,
                uint64 mask);
        int bcm_field_qualify_NativeVSwitch(
        int unit,
        bcm_field_entry_t entry,
        bcm_vlan_t data,
        bcm_vlan_t mask);
        int bcm_field_qualify_NativeVSwitchValid(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 mask);
        int bcm_field_qualify_TrillIVersion(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 mask);
        int bcm_field_qualify_TrillIngressRbridge(
        int unit,
        bcm_field_entry_t entry,
        uint16 data,
        uint16 mask);
        int bcm_field_qualify_TrillMultiDestination(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 mask);
        int bcm_field_qualify_TrillOpLength(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 mask);
        int bcm_field_qualify_TrillHopCount(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 mask);
        int bcm_field_qualify_TrillPayload(
        int unit,
        bcm_field_entry_t entry,
        uint64 data,
        uint64 mask);
        int bcm_field_qualify_MplsForwardingLabelPayload(
        int unit,
        bcm_field_entry_t entry,
        uint64 data,
        uint64 mask);
        int bcm_field_qualify_IncomingIpIfClass(
        int unit,
        bcm_field_entry_t entry,
        uint32 data,
        uint32 mask);
        int bcm_field_qualify_L2GreProtocolType(
        int unit,
        bcm_field_entry_t entry,
        uint16 data,
        uint16 mask);
        int bcm_field_qualify_L2GreVsid(
        int unit,
        bcm_field_entry_t entry,
        uint32 data,
        uint32 mask);
        int bcm_field_qualify_L2GreFlowId(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 mask);
        int bcm_field_qualify_MimVlanPri(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 mask);
        int bcm_field_qualify_MimVlanCfi(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 mask);
        int bcm_field_qualify_Fhei(
                int unit,
                bcm_field_entry_t entry,
                uint64 data,
                uint64 mask);
        int bcm_field_qualify_FheiSize(
                int unit,
                bcm_field_entry_t entry,
                uint32 data,
                uint32 mask);
        int bcm_field_qualify_udf(
                int unit,
                bcm_field_entry_t eid,
                bcm_udf_id_t udf_id,
                int length,
                uint8 *data,
                uint8 *mask);
 int bcm_field_qualify_OuterVlanActionRange(
          int unit,
        bcm_field_entry_t entry,
        bcm_vlan_t data,
        bcm_vlan_t mask);
 int bcm_field_qualify_InnerVlanActionRange(
        int unit,
        bcm_field_entry_t entry,
        bcm_vlan_t data,
        bcm_vlan_t mask);
 int bcm_field_qualify_ClassId(
        int unit,
        bcm_field_entry_t entry,
        uint8 data,
        uint8 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_E_xxx

bcm_field_qualify_SrcNivGport

Add SrcNivGport qualification to a field entry

Syntax

```
       int bcm_field_qualify_SrcNivGport(
               int unit,
               bcm_field_entry_t entry,
               bcm_gport_t niv_port_id);
```

Parameters

unit	BCM device number
entry	Field entry ID
niv_port_id	Data to match against

Description

Set match criteria for SrcNivGport qualifier

Returns

`BCM_E_xxx`

bcm_field_qualify_DstNivGport

Add DstNivGport qualification to a field entry

Syntax

```
       int bcm_field_qualify_DstNivGport(
               int unit,
               bcm_field_entry_t entry,
               bcm_gport_t niv_port_id);
```

Parameters

`unit`	BCM device number
`entry`	Field entry ID
`niv_port_id`	Data to match against

Description

Set match criteria for DstNivGport qualifier

Returns

`BCM_E_xxx`

bcm_field_qualify_DstGport

Add DstGport qualification to a field entry

Syntax

```
       int bcm_field_qualify_DstGport(
               int unit,
               bcm_field_entry_t entry,
               bcm_gport_t port_id);
```

Parameters

`unit`	BCM device number
`entry`	Field entry ID
`port_id`	Data to match against

Description

Set match criteria for DstGport qualifier

Returns

BCM_E_xxx

bcm_field_qualify_XXX_get

Get a qualification match criteria from a field entry

Syntax

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

int bcm_field_qualify_IsEqualValue_get(
        int unit,
        bcm_field_entry_t entry,
        uint8 *data,
        uint8 *mask);
int bcm_field_qualify_MirrorEgressDisabled_get(
int unit,
bcm_field_entry_t entry,
uint8 *data,
uint8 *mask);
int bcm_field_qualify_BypassSrcMacFilter_get(
int unit,
bcm_field_entry_t entry,
uint8 *data,
uint8 *mask);
int bcm_field_qualify_RxTrapCodeForSnoop_get(
int unit,
bcm_field_entry_t entry,
uint32 *data,
uint32 *mask);
int bcm_field_qualify_IpmcStarGroupValue_get(
        int unit,
        bcm_field_entry_t entry,
        uint64 *data,
        uint64 *mask);
int bcm_field_qualify_NativeVSwitch_get(
int unit,
bcm_field_entry_t entry,
bcm_vlan_t *data,
bcm_vlan_t *mask);
int bcm_field_qualify_NativeVSwitchValid_get(
int unit,
bcm_field_entry_t entry,
uint8 *data,
uint8 *mask);
int bcm_field_qualify_TrillIVersion_get(
int unit,
bcm_field_entry_t entry,
uint8 *data,
uint8 *mask);
int bcm_field_qualify_TrillIngressRbridge_get(
int unit,
bcm_field_entry_t entry,
uint16 *data,
uint16 *mask);
int bcm_field_qualify_TrillMultiDestination_get(
int unit,
bcm_field_entry_t entry,
uint8 *data,
uint8 *mask);
int bcm_field_qualify_TrillOpLength_get(
int unit,
bcm_field_entry_t entry,
uint8 *data,
uint8 *mask);
int bcm_field_qualify_TrillHopCount_get(
int unit,
bcm_field_entry_t entry,
uint8 *data,
uint8 *mask);
int bcm_field_qualify_TrillPayload_get(
int unit,
bcm_field_entry_t entry,
uint64 *data,
uint64 *mask);
int bcm_field_qualify_MplsForwardingLabelPayload_get(
int unit,
bcm_field_entry_t entry,
uint64 *data,
uint64 *mask);
int bcm_field_qualify_IncomingIpIfClass_get(
int unit,
bcm_field_entry_t entry,
uint32 *data,
uint32 *mask);
int bcm_field_qualify_L2GreProtocolType_get(
int unit,
bcm_field_entry_t entry,
uint16 *data,
uint16 *mask);
int bcm_field_qualify_L2GreVsid_get(
int unit,
bcm_field_entry_t entry,
uint32 *data,
uint32 *mask);
int bcm_field_qualify_L2GreFlowId_get(
int unit,
bcm_field_entry_t entry,
uint8 *data,
uint8 *mask);
int bcm_field_qualify_MimVlanPri_get(
int unit,
bcm_field_entry_t entry,
uint8 *data,
       uint8 *mask);
       int bcm_field_qualify_MimVlanCfi_get(
       int unit,
       bcm_field_entry_t entry,
       uint8 *data,
       uint8 *mask);
       int bcm_field_qualify_Fhei_get(
               int unit,
               bcm_field_entry_t entry,
               uint64 *data,
               uint64 *mask);
       int bcm_field_qualify_FheiSize_get(
               int unit,
               bcm_field_entry_t entry,
               uint32 *data,
               uint32 *mask);
       int bcm_field_qualify_udf_get(
               int unit,
               bcm_field_entry_t eid,
               bcm_udf_id_t udf_id,
               int max_length,
               uint8 *data,
               uint8 *mask,
               int *actual_length);
int bcm_field_qualify_OuterVlanActionRange_get(
          int unit,
       bcm_field_entry_t entry,
       bcm_vlan_t *data,
       bcm_vlan_t *mask);
int bcm_field_qualify_InnerVlanActionRange_get(
       int unit,
       bcm_field_entry_t entry,
       bcm_vlan_t *data,
       bcm_vlan_t *mask);
int bcm_field_qualify_ClassId_get(
       int unit,
       bcm_field_entry_t entry,
       uint8 *data,
       uint8 *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.

Returns

BCM_E_xxx

bcm_field_qualify_SrcNivGport_get

Get the SrcNivGport qualification match criteria from a field entry

Syntax

```
       int bcm_field_qualify_SrcNivGport_get(
               int unit,
               bcm_field_entry_t entry,
               bcm_gport_t *niv_port_id);
```

Parameters

unit	BCM device number
entry	Field entry ID
niv_port_id	Data to match against

Description

Get match criteria for SrcNivGport qualifier from a field entry.

Returns

BCM_E_xxx

bcm_field_qualify_DstNivGport_get

Get the DstNivGport qualification match criteria from a field entry

Syntax

```
       int bcm_field_qualify_DstNivGport_get(
               int unit,
               bcm_field_entry_t entry,
               bcm_gport_t *niv_port_id);
```

Parameters

unit	BCM device number
entry	Field entry ID
niv_port_id	Data to match against

Description

Get match criteria for DstNivGport qualifier from a field entry.

Returns

```
BCM_E_xxx
```

bcm_field_qualify_DstGport_get

Get the DstGport qualification match criteria from a field entry

Syntax

```
       int bcm_field_qualify_DstGport_get(
               int unit,
               bcm_field_entry_t entry,
               bcm_gport_t *port_id);
```

Parameters

unit	BCM device number
entry	Field entry ID
port_id	Data to match against

Description

Get match criteria for DstGport qualifier from a field entry.

Returns

```
BCM_E_xxx
```

bcm_field_stage_info_get

Get per-stage field information

Syntax

```
#include <bcm/field.h>
int bcm_field_stage_info_get(
    int unit,
    bcm_field_stage_t stage,
    bcm_field_stage_info_t *info);
```

Parameters

unit	(IN) Unit number
stage	(IN) field stage
stage	(OUT) stage info

Description

per stage information, such as preselectors managemnt advanced mode, maximum preselecrots ID etc.

Returns

`BCM_E_xxx`

`BCM_E_INIT`	Field module not initialized
`BCM_E_XXX`	Other error code
`BCM_E_NONE`	Success

bcm_field_stat_sync_get

Force an immediate counter update and retrieve specific statistic type.

Syntax

```
#include <bcm/field.h>
int bcm_field_stat_sync_get(int unit, int stat_id,
    bcm_field_stat_t  stat, uint64 *value);
```

Parameters

`unit`	(IN) BCM device number
`stat_id`	(IN) Statistics entity ID.
`stat`	(IN) Specific statistics type.
`value`	(OUT) Counter value for statistics type.

Description

Similar to `bcm_field_stat_get()`, value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_XXX`

bcm_field_stat_sync_get32

Force an immediate counter update and retrieve specific statistic type.
Get 32-bit counter value for specific statistic type.

Syntax

```
#include <bcm/field.h>
int bcm_field_stat_sync_get32(int unit, int stat_id,
      bcm_field_stat_t  stat, uint32 *value);
```

Parameters

`unit`	(IN) BCM device number
`stat_id`	(IN) Statistics entity ID.
`stat`	(IN) Specific statistics type.
`value`	(OUT) Counter value for statistics type.

Description

Similar to `bcm_field_stat_get32()`, value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_XXX
```

bcm_field_stat_counter_sync_get

Force an immediate counter update and retrieve statistics value for a field entity

Syntax

```
#include <bcm/field.h>
int
bcm_field_stat_counter_sync_get(
    int                unit,
    uint32             stat_id,
    bcm_field_stat_t stat,
    uint32             num_entries,
    uint32             *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

`unit`	(IN) Unit number.
`stat_id`	(IN) Statistics entity ID
`stat`	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_field_stat_counter_get()`, value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_xxx
```

IP MULTICAST

New APIs `bcm_ipmc_stat_counter_sync_get()` is added in this release.

bcm_ipmc_stat_counter_sync_get

Force an immediate counter update and retrieve counter statistic values
for an IPMC group.

Syntax

```
#include <bcm/ipmc.h>
int
bcm_ipmc_stat_counter_sync_get(
    int                unit,
    bcm_ipmc_addr_t    *info,
    bcm_ipmc_stat_t    stat,
    uint32             num_entries,
    uint32             *counter_indexes,
    bcm_stat_value_t   *counter_values);
```

Parameters

`unit`	(IN) Unit number
`info`	(IN) IPMC entry information
`stat`	(IN) Type of the counter to retrieve(I.e. ingress/egress byte/packet)
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_ipmc_stat_counter_get()`, value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_xxx
```

LAYER 2 ADDRESS MANAGEMENT

New control flag `BCM_L2_EGRESS_VLAN_QOS_MAP_REPLACE` has been added for `bcm_l2_egress_t` data structure.

Table 30: BCM L2 Egress Flags

Name	Purpose
BCM_L2_EGRESS_VLAN_QOS_MAP_REPLACE	Replace VLAN PCP-DEI QOS settings

New flag field BCM_L2_CACHE_PROTO_PKT has been added for bcm_l2_cache_addr_t.

Table 31: BCM L2 Cache Flags

Name	Purpose
BCM_L2_CACHE_PROTO_PKT	Mark the frame as protocol packet.

New replace flag BCM_L2_REPLACE_MATCH_UC and BCM_L2_REPLACE_MATCH_MC have been added for bcm_l2_replace() API.

Table 32: L2 Replace Flags

Name	Purpose
BCM_L2_REPLACE_MATCH_UC	L2 replace will be performed on Unicast entries
BCM_L2_REPLACE_MATCH_MC	L2 replace will be performed on Multicast entries

The field name of L2 multicast group index has been changed from l2mc_index to l2mc_group.

```
typedef struct bcm_l2_addr_s {
    ...
    bcm_cos_t           cos_src;         /* CoS based on src addr */
    bcm_multicast_t     l2mc_group;      /* XGS: index in L2MC table */
    bcm_pbmp_t          block_bitmap;    /* XGS: blocked egress bitmap */
    ...
} bcm_l2_addr_t;
```

New fields of bcm_l2_egress_t data structure have been added.

```
/* Device-independent L2 egress structure. */
typedef struct bcm_l2_egress_s {
    ...
    uint16 inner_tpid;            /* inner vlan tag TPID */
    bcm_if_t l3_intf;             /* L3 interface (tunnel). Used only in
                                 Native Routing overlay protocols. */

    int vlan_qos_map_id;          /* VLAN PCP-DEI QoS settings. Used only
                                 in Native Routing overlay protocols. */
} bcm_l2_egress_t;
```

L2GRE MANAGEMENT

BCM_L2GRE_PORT_DEFAULT was incorrectly added in the enumeration bcm_l2gre_port_match_t for MPLS Port Match Criteria. This has now been corrected.

Table 33: L2GRE port flags

Name	***Purpose***
`BCM_L2GRE_PORT_DEFAULT`	Create Multicast-only L2GRE Tunnel

New API `bcm_l2gre_stat_counter_sync_get()` has been added in this release.

bcm_l2gre_stat_counter_sync_get

Force an immediate counter update and retrieve counter statistic values for
a l2gre vp/vpn index.

Syntax

```
#include <bcm/l2gre.h>
int
bcm_l2gre_stat_counter_sync_get(
    int                unit,
    bcm_gport_t       port,
    bcm_vpn_t         vpn,
    bcm_l2gre_stat_t    stat,
    uint32             num_entries,
    uint32             *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

`unit`	(IN) Unit number.
`port`	(IN) l2gre vp index
`vpn`	(IN) l2gre vpn index
`stat`	(IN) l2gre counter stat types.
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_l2gre_stat_counter_get()`, value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_xxx`

LAYER 3 MANAGEMENT

A new field `native_routing_vlan_tags` of `bcm_l3_intf_t` for L3 interface structure has been added in this release.

```
typedef struct bcm_l3_intf_s {
    ...
    uint8 native_routing_vlan_tags; /* Set number of VLAN tags expected
                                     * when interface is used for native
                                     * routing
                                     */

} bcm_l3_intf_t;
```

A new flag `BCM_L3_ECMP_PATH_NO_SORTING` for L3 ECMP flags has been added.

Table 34: BCM L3 ECMP Flags

Name	***Purpose***
`BCM_L3_ECMP_PATH_NO_SORTING`	If set, the members of the ECMP group won't be resorted.

A new flag `BCM_L3_SOURCE_BIND_USE_MASK` for L3 Source Binding flags as well as the new IPv4 and IPv6 subnet masks for `bcm_l3_source_bind_t` data structure have been added in this release.

Table 35: BCM Source Binding Flags

Name	***Purpose***
`BCM_L3_SOURCE_BIND_USE_MASK`	Indicate mask fields are to be used

```
typedef struct bcm_l3_source_bind_s {
    ...
    bcm_ip_t ip_mask;               /* IP subnet mask (IPv4). */
    bcm_ip6_t ip6_mask;             /* IP subnet mask (IPv6). */
} bcm_l3_source_bind_t;

typedef bcm_l3_source_bind_t bcmx_l3_source_bind_t;
```

A new flag `BCM_TUNNEL_WITH_ID` has been added in this release

Table 36: BCM Tunnel Flags

Name	***Purpose***
`BCM_TUNNEL_WITH_ID`	Add using the specified ID.

New L3 APIs `bcm_l3_xxx_stat_sync_get/get32()` and `bcm_l3_xxx_stat_counter_sync_get/get32()` APIs as well as the `bcm_l3_route_find()` and `bcm_l3_subnet_route_find()` have been added in this release.

bcm_l3_nat_egress_stat_counter_sync_get

Force an immediate counter update and retrieve counter statistic values for a Egress NAT Index.

Syntax

```
#include <bcm/nat.h>
int
bcm_l3_nat_egress_stat_counter_sync_get(
    int                 unit,
    bcm_l3_nat_egress_t    *info,
    bcm_l3_nat_egress_stat_t    stat,
    uint32              num_entries,
    uint32              *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

`unit`	(IN) Unit number.
`info`	(IN) Egress NAT Index
`stat`	(IN) Egress Nat Counter stat types.
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_l3_nat_egress_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_xxx`

bcm_l3_vrf_stat_sync_get
bcm_l3_vrf_stat_sync_get32

Force an immediate counter update and retrieve L3 VRF counter value for specified VRF statistic type.

Syntax

```
#include <bcm/l3.h>
int
bcm_l3_vrf_stat_sync_get(int unit, bcm_vrf_t vrf, bcm_l3_vrf_stat_t stat,
    uint64 *val);

int
bcm_l3_vrf_stat_sync_get32(int unit, bcm_vrf_t vrf, bcm_l3_vrf_stat_t stat,
      uint32 *val);
```

Parameters

`unit`	(IN) Unit number.
`vrf`	(IN) Virtual router instance.
`stat`	(IN) Type of the counter to retrieve.
`val`	(OUT) Pointer to a counter value.

Description

Similar to `bcm_l3_vrf_stat_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_XXX`

bcm_l3_vrf_stat_counter_sync_get

Force an immediate counter update and retrieve L3 VRF counter value for
specified VRF statistic type

Syntax

```
#include <bcm/l3.h>
int
bcm_l3_vrf_stat_counter_sync_get(
    int               unit,
    bcm_vrf_t         vrf,
    bcm_l3_vrf_stat_t stat,
    uint32            num_entries,
    uint32            *counter_indexes,
    bcm_stat_value_t  *counter_values);
```

Parameters

`unit`	(IN) Unit number.
`vrf`	(IN) Virtual router instance
`stat`	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_l3_vrf_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter

Returns

`BCM_E_xxx`

bcm_l3_egress_stat_counter_sync_get

Force an immediate counter update and retrieve the specified counter statistic for a L3 egress interface.

Syntax

```
#include <bcm/l3.h>
int
bcm_l3_egress_stat_counter_sync_get(
    int              unit,
    bcm_if_t         intf_id,
    bcm_l3_stat_t    stat,
    uint32           num_entries,
    uint32           *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

unit	(IN) Unit number.
intf_id	(IN) Interface ID of a egress L3 object.
stat	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
num_entries	(IN) Number of counter Entries
counter_indexes	(IN) Pointer to Counter indexes entries
counter_values	(OUT) Pointer to counter values

Description

Similar to `bcm_l3_egress_stat_counter_get()`, value returned is software accumulated counter synced with the hardware counter

Returns

`BCM_E_xxx`

bcm_l3_ingress_stat_counter_sync_get

Get counter statistic values for a l3 interface object.

Syntax

```
#include <bcm/l3.h>
int
bcm_l3_ingress_stat_counter_sync_get(
    int              unit,
    bcm_if_t         intf_id,
    bcm_l3_stat_t    stat,
    uint32           num_entries,
    uint32           *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

`unit`	(IN) Unit number.
`intf_id`	(IN) Interface ID of a L3 ingress object
`stat`	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_l3_ingress_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_xxx`

bcm_l3_host_stat_counter_sync_get

Force an immediate counter update and retrieve the specified counter
statistic for a L3 host entry.

Syntax

```
#include <bcm/l3.h>
int bcm_l3_host_stat_counter_sync_get(
    int unit,
    bcm_l3_host_t *info,
    bcm_l3_stat_t stat,
    uint32 num_entries,
    uint32 *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

`unit`	(IN) unit number
`info`	(IN) L3 host description
`stat`	(IN) Type of the counter to retrieve
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_l3_host_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_XXX
```

bcm_l3_route_stat_counter_sync_get

Force an immediate counter update and retrieve counter statistic values for a l3 route index.

Syntax

```
#include <bcm/l3.h>
int
bcm_l3_route_stat_counter_sync_get(
    int                 unit,
    bcm_l3_route_t     *info,
    bcm_l3_route_stat_t     stat,
    uint32              num_entries,
    uint32              *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

unit	(IN) Unit number.
info	(IN) L3 Route Index
stat	(IN) L3 route counter stat types.
num_entries	(IN) Number of counter Entries
counter_indexes	(IN) Pointer to Counter indexes entries
counter_values	(OUT) Pointer to counter values

Description

Similar to `bcm_l3_route_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_xxx
```

bcm_l3_route_find

Find a longest prefix matched route given an IP address.

Syntax

```
#include <bcm/l3.h>
int bcm_l3_route_find(int unit, bcm_l3_host_t *host, bcm_l3_route_t *route);
```

Parameters

`unit`	BCM device number
`host`	Pointer to bcm_l3_host_t specifying the IP address.
`route`	Pointer to bcm_l3_route_t specifying the network number.

Description

Gets longest prefix matched route for a host from the route table(Support for ALPM mode only). Match conditions: for each entry in the route tables, compare input IP address & entry mask with entry subnet & entry mask, if they are equal, than pickup the longest prefix from the matched entries. The return only indicates the result of route table process, not represent the result of the chip process.

The valid flags are as follows:

- `BCM_L3_MULTIPATH` - Set to designate the route is ECMP route. In this case, only the information for the first path is returned.
- `BCM_L3_IP6` - Address is IPv6.
- `BCM_L3_HIT_CLEAR` - Clear HIT bit if set

Table 37: L3 Route Find Fields

Fields	***Description***
`l3a_flags`	Operation control flags
`l3a_vrf`	Virtual router instance, if applicable
`l3a_net`	Network IP address
`l3a_ipmask`	Network IP mask
`l3a_intf`	(OUT) L3 interface
`l3a_nexthopip`	(OUT) Next hop IP address (gateway) - StrataXGS I/II only
`l3a_nexthop_mac`	(OUT) Next hop MAC address
`l3a_port_tgid`	(OUT) Port or trunk group ID
`l3a_modid`	(OUT) Module ID
`l3a_tunnel_option`	(OUT) reused for memory table index
`l3a_vid`	(OUT) VLAN ID for per-VLAN default route (BCM5695 only)

Returns

`BCM_E_NOT_FOUND`	- Routes not found
`BCM_E_NONE`	- Found a matched route
`BCM_E_XXX`	- Other error code

l3_subnet_route_find

Find a longest prefix matched route given the ip and netmask.

Syntax

```
#include <bcm/l3.h>
```

```
int bcm_l3_subnet_route_find(int unit, bcm_l3_route_t *input, bcm_l3_route_t
*route);
```

Parameters

`unit`	BCM device number
`input`	Pointer to bcm_l3_route_t specifying the subnet.
`route`	Pointer to bcm_l3_route_t specifying the network number.

Description

Gets information for an IP route from the route table(Support for ALPM mode only). Match conditions: for each entry in the route tables, entry prefix length <= input prefix length, compare input subnet & entry mask with entry subnet & entry mask, if they are equal, than pickup the longest prefix from the matched entries. The return only indicates the result of route table process, not represent the result of the chip process.

The valid flags are as follows:

- `BCM_L3_MULTIPATH` - Set to designate the route is ECMP route. In this case, only the information for the first path is returned.
- `BCM_L3_IP6` - Address is IPv6.
- `BCM_L3_HIT_CLEAR` - Clear HIT bit if set

Table 38: L3 Subnet Route Find Fields

Fields	***Description***
`l3a_flags`	Operation control flags
`l3a_vrf`	Virtual router instance, if applicable
`l3a_net`	Network IP address
`l3a_ipmask`	Network IP mask
`l3a_intf`	(OUT) L3 interface
`l3a_nexthopip`	(OUT) Next hop IP address (gateway) - StrataXGS I/II only
`l3a_nexthop_mac`	(OUT) Next hop MAC address
`l3a_port_tgid`	(OUT) Port or trunk group ID
`l3a_modid`	(OUT) Module ID
`l3a_tunnel_option`	(OUT) reused for memory table index
`l3a_vid`	(OUT) VLAN ID for per-VLAN default route (BCM5695 only)

Returns

`BCM_E_NOT_FOUND`	- Routes not found
`BCM_E_NONE`	- Found a matched route
`BCM_E_XXX`	- Other error code

MAC-IN-MAC MANAGEMENT

`BCM_MIM_VPN_ELINE` is newly defined for Point-to-Point VPN service for MAC-in-MAC VPN Flags.

Table 39: MAC-in-MAC VPN Config Flags

Name	*Purpose*
BCM_MIM_VPN_ELINE	Must be set to specify that the VPN type is Point-to-Point(E-Line) MAC-in-MAC.

New APIs bcm_mim_lookup_id_stat_counter_sync_get() and bcm_mim_vpn_stat_counter_sync_get() are added in this release.

bcm_mim_lookup_id_stat_counter_sync_get

Force an immediate counter update and retrieve counter statistic values
for specific MiM I-SID entry.

Syntax

```
#include <bcm/mim.h>
int bcm_mim_lookup_id_stat_counter_sync_get(
    int unit,
    int lookup_id,
    bcm_mim_stat_t stat,
    uint32 num_entries,
    uint32 *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

unit	(IN) unit number
lookup_id	(IN) I-SID value
stat	(IN) Type of the counter to retrieve
num_entries	(IN) Number of counter Entries
counter_indexes	(IN) Pointer to Counter indexes entries
counter_values	(OUT) Pointer to counter values

Description

Similar to bcm_mim_lookup_id_stat_counter_get(), value returned is software accumulated counter synced with the hardware counter.

Returns

BCM_E_XXX

bcm_mim_vpn_stat_counter_sync_get

Force an immediate counter update and retrieve counter statistic values for
specific MiM VPN entry

Syntax

```
#include <bcm/mim.h>
int bcm_mim_vpn_stat_counter_sync_get(
    int unit,
    bcm_mim_vpn_t vpn,
    bcm_mim_stat_t stat,
    uint32 num_entries,
    uint32 *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

`unit`	(IN) unit number
`vpn`	(IN) MiM VPN
`stat`	(IN) Type of the counter to retrieve
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_mim_vpn_stat_counter_get()`, value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_XXX`

MIRRORING

A new flag `BCM_MIRROR_DEST_TUNNEL_ETAG` for mirror destination flags as well as the required fields of `bcm_mirror_destination_t` data structure has been added.

Table 40: BCM Mirror Destination Flags

Name	*Description*
`BCM_MIRROR_DEST_TUNNEL_ETAG`	Mirrored packet should be ETAG(Port Extender VLAN Tag) tunneled. May be combined with L2 or L3 GRE tunnels.

```
struct bcm_mirror_destination_s {
    ...
    uint16 etag_src_vid;                /* Extended (source) port vlan id */
   uint16 etag_dst_vid;              /* Extended (destination) port vlan id */
   uint32 egress_sample_rate_dividend; /* The probability of outbound mirroring
                       a packet from the destination is
                       sample_rate_dividend >=
                       sample_rate_divisor ? 1 :
```

```
                       sample_rate_dividend /
                       sample_rate_divisor */
    uint32 egress_sample_rate_divisor;
    uint8 recycle_context;              /* recycle context of egress originating
                       packets */
    uint16 packet_copy_size;             /*  If non zero, the copied packet will
                       be truncated to the first
                       packet_copy_size . Current supported
                       values for DNX are 0, 64, 128, 192 */
    uint16 egress_packet_copy_size;     /* If non zero and the packet is copied
                       from the egress, the packet will be
                       truncated to the first
                       packet_copy_size . Current supported
                       values for DNX are 0, 256. */
    bcm_mirror_pkt_header_updates_t packet_control_updates;
} bcm_mirror_destination_t;

typedef bcm_mirror_destination_t bcmx_mirror_destination_t;
```

New mirroring APIs have been added with the corresponding data types.

```
/*  represents the options for the mirroring of packets */
typedef struct bcm_mirror_options_s {
    uint32 flags;
    uint8 forward_strength;
    uint8 copy_strength;
} bcm_mirror_options_t;

/*   represents header updates of mirrored packets */
typedef struct bcm_mirror_pkt_header_updates_s {
   uint32 valid;                    /* Used to specify which fields to use for
                      header changes. Possible values will be
                      named BCM_MIRROR_PKT_HEADER_UPDATE_* */
   bcm_color_t color;               /* drop precedence */
   uint8 prio;                      /* the internal packet priority (traffic
                      class before ingress mapping to cosq) */
   uint8 ecn_value;                 /* ECN capable and congestion encoding */
   uint8 cnm_cancel;                /* Ignore Congestion Point (CNM) */
   uint32 trunk_hash_result;        /* LAG load balancing key */
   bcm_gport_t in_port;             /* 8b should be exposed? */
   uint16 vsq;                      /* selects STF (statistics flow) VSQ */
   uint16 fabric_header_editing;    /* changes to the fabric headers */
} bcm_mirror_pkt_header_updates_t;
```

bcm_mirror_pkt_header_updates_t_init

Initialized bcm_mirror_pkt_header_updates_t struct

Syntax

```
#include <bcm/mirror.h>
void bcm_mirror_pkt_header_updates_t_init(bcm_mirror_pkt_header_updates_t
*updates);
```

Parameters

`updates` (OUT) mirror pkt header_updates to initialized.

Description

Initialize a mirror pkt `header_updates` structure.

Returns

None.

bcm_mirror_options_t_init
bcm_mirror_port_destination_add
bcm_mirror_port_destination_get
bcm_mirror_port_vlan_destination_add
bcm_mirror_port_vlan_destination_get

extended versions of existing _dest_ APIs using a bcm_mirror_options_t argument

Syntax

```
#include <bcm/mirror.h>
        void bcm_mirror_options_t_init(bcm_mirror_options_t *options);
int bcm_mirror_port_destination_add(int unit, bcm_port_t port,
     uint32 flags, bcm_gport_t mirror_dest,bcm_mirror_options_t options);

int bcm_mirror_port_destination_get(int unit, bcm_port_t port, uint32 flags,
     int mirror_dest_size, bcm_gport_t *mirror_dest,
     int *mirror_dest_count,bcm_mirror_options_t *options);

int bcm_mirror_port_vlan_dest_add(int unit, bcm_port_t port, bcm_vlan_t vlan,
uint32 flags, bcm_gport_t destid, bcm_mirror_options_t options);

int bcm_mirror_port_vlan_dest_get(int unit, bcm_port_t port, bcm_vlan_t vlan,
uint32 flags,unsigned int mirror_dest_size, bcm_gport_t *destid,unsigned int
*destcount,bcm_mirror_options_t *options);
```

Parameters

`options` (IN|OUT) The options for the mirroring of packets(input for *_add api's,output for *_get api's)

Description

extended versions of existing `_dest_` APIs using a `bcm_mirror_options_t` argument

Returns

```
BCM_E_XXX
```

MPLS MANAGEMENT

A new MPLS port flag, `BCM_MPLS_PORT_FORWARD_GROUP,` has been added as well as the extension of MPLS port flags.

Table 41: MPLS Port Flags

Name	Purpose
BCM_MPLS_PORT_FORWARD_GROUP	Enable VLAN-Port property FORWARD_GROUP to MPLS-Port.

Table 42: MPLS Port Flags(2)

Name	Purpose
BCM_MPLS_PORT2_INGRESS_ONLY	Indicates ingress settings
BCM_MPLS_PORT2_EGRESS_ONLY	Indicates egress settings
BCM_MPLS_PORT2_EGRESS_PROTECTION	Egress protection object

Two new fields of MPLS port type `bcm_mpls_port_t` have been added regarding failover stuff.

```
/* MPLS port type. */
typedef struct bcm_mpls_port_s {
    ...
    bcm_failover_t egress_failover_id;  /* Failover object index for
                       Egress Protection */
    bcm_gport_t egress_failover_port_id;/* Failover MPLS Port identifier
                                           for Egress Protection */
} bcm_mpls_port_t;
```

New APIs `bcm_mpls_port_stat_counter_sync_get()` and `bcm_mpls_label_stat_counter_sync_get()` as well as `bcm_mpls_label_stat_sync_get/get32()` APIs are added in this release.

bcm_mpls_port_stat_counter_sync_get

Force an immediate counter update and retrieve get counter statistic
values for specific vpn and gport

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_port_stat_counter_sync_get(
    int                   unit,
    bcm_vpn_t             vpn,
    bcm_gport_t           port,
    bcm_mpls_port_stat_t stat,
    uint32                num_entries,
    uint32                *counter_indexes,
    bcm_stat_value_t      *counter_values);
```

Parameters

unit	(IN) Unit number.
vpn	(IN) VPN ID
port	(IN) MPLS Gport
stat	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
num_entries	(IN) Number of counter Entries
counter_indexes	(IN) Pointer to Counter indexes entries
counter_values	(OUT) Pointer to counter values

Description

Similar to `bcm_mpls_port_stat_counter_sync_get()`, value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_xxx`

bcm_mpls_label_stat_counter_sync_get

Force an immediate counter update and retrieve counter statistic values for
specific MPLS label and gport

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_label_stat_counter_sync_get(
    int                   unit,
    bcm_mpls_label_t      label,
    bcm_gport_t           port,
    bcm_mpls_port_stat_t stat,
    uint32                num_entries,
    uint32                *counter_indexes,
    bcm_stat_value_t      *counter_values);
```

Parameters

`unit`	(IN) Unit number.
`label`	(IN) MPLS Label
`port`	(IN) MPLS Gport
`stat`	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_mpls_label_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_xxx`

bcm_mpls_label_stat_sync_get bcm_mpls_label_stat_sync_get32

Force an immediate counter update and retrieve MPLS Stats.

Syntax

```
#include <bcm/mpls.h>
int bcm_mpls_label_stat_sync_get(int unit, bcm_mpls_label_t label,
 bcm_gport_t port, bcm_mpls_stat_t stat,
 uint64 *val);

int bcm_mpls_label_stat_sync_get32(int unit, bcm_mpls_label_t label,
   bcm_gport_t port, bcm_mpls_stat_t stat,
   uint32 *val);
```

Parameters

`unit`	(IN) BCM device number
`label`	(IN) MPLS Label
`port`	(IN) Gport
`stat`	(IN) Specify the Stat Type
`val`	(OUT) Pointer to stats value

Description

Similar to `bcm_mpls_label_stat_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_XXX`

MULTICAST CONFIGURATION

New mulaticast APIs have been added for this release.

bcm_multicast_set
bcm_multicast_get

Set or Get all the replicataions of a multicast group.

Syntax

```
#include <bcm/multicast.h>
int bcm_multicast_set(
    int unit,
    bcm_multicast_t group,
    uint32 flags,
    int nof_replications,
    bcm_multicast_replication_t *rep_array);
int bcm_multicast_get(
    int unit,
    bcm_multicast_t group,
    uint32 flags,
    int replication_max,
    bcm_multicast_replication_t *out_rep_array,
    int *rep_count);
```

Parameters

`unit`	(IN) Unit number.
`group`	(IN) Multicast group ID
`flags`	(IN) flags for the API
`nof_replications`	(IN) number of replications to be set in the group
`rep_array`	(IN) the replications to be set in the group
`replication_max`	(IN) max number of replications to be returned (size of rep_array)
`out_rep_array`	(OUT) Array of replications to hold the output replications
`rep_count`	(OUT) Actual number of replications returned in the array

Description

Set/Get the replications of the given multicast group. Set will disregard any existing replications.

Table 43: values for the flag parameter of the new api family bcm_multicast_get/set/add/delete

Name	*Purpose*
`BCM_MULTICAST_INGRESS`	Determines if the operation is on an ingress multicast group. If not specified an egress multicast group is assumed.

The purpuse of `bcm_multicast_replication_t` is to hold all the information of one replication.

```
typedef struct {
    uint32 flags;     /* information on the replication */
    bcm_gport_t port; /* replication destination */
    bcm_if_t encap1;  /* first encapsulation */
    bcm_if_t encap2;  /* second encapsulation */
} bcm_mullticast_replication_t;  /* represents a multicast replication */
```

Table 44: bcm_mullticast_replication_t Flags

Name	*Purpose*
`BCM_MUTICAST_REPLICATION_ENCAP2_VALID`	determines if encap2 is used
`BCM_MUTICAST_REPLICATION_ENCAP1_L3_INTF`	when encap2 is used, determines if encap1 is a routing interface or not

Returns

`BCM_E_XXX`

bcm_multicast_delete
bcm_multicast_add

Delete or add replications of a multicast group.

Syntax

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

int bcm_multicast_delete(
    int unit,
    bcm_multicast_t group,
    uint32 flags,
    int nof_replications,
    bcm_multicast_replication_t *rep_array);
int bcm_multicast_add(
    int unit,
    bcm_multicast_t group,
    uint32 flags,
    int nof_replications,
    bcm_multicast_replication_t *rep_array);
```

Parameters

`unit`	(IN) Unit number.
`group`	(IN) Multicast group ID
`flags`	(IN) flags for the API
`nof_replications`	(IN) number of replications to delete/add
`rep_array`	(IN) the replications to be deleted/added

Description

delete or add replications from/to a given multicast group.

Returns

BCM_E_XXX

multicast_replication_t_init

Init the multicast_replication_t structure.

Syntax

```
#include <bcm/multicast.h>
 bcm_multicast_replication_t_init(
    bcm_multicast_replication_t *replication_structure);
```

Parameters

`replication_structu` `re`	(OUT) replication structure for initialization

NETWORK INTERFACE VIRTUALIZATION MANAGEMENT

A new NIV port flag has been added for the flags field of bcm_niv_port_t structure that describes a NIV virtual port used in VIS.

```
typedef struct bcm_niv_port_s {
    uint32 flags;                   /* BCM_NIV_PORT_xxx */
    bcm_gport_t niv_port_id;        /* BCM_GPORT_NIV_PORT type object */
    bcm_gport_t port;               /* Physical port / trunk */
    uint16 virtual_interface_id;    /* Virtual interface identifier */
    bcm_vlan_t match_vlan;          /* Optional outer VLAN ID to match. */
    uint16 match_service_tpid;      /* Ingress SD-tag TPID. */
    uint32 if_class;                /* Interface class ID. */
} bcm_niv_port_t;
```

Table 45: NIV port flags

Name	*Purpose*
BCM_NIV_VNTAG_L_BIT_FORCE_1	Allow frames head back

New fields intf and multicast_flags of bcm_niv_egress_t data structure have been added along with their NIV egress type of BCM_NIV_EGRESS_L3.

```
/* NIV egress type. */
typedef struct bcm_niv_egress_s {
    ...
    bcm_if_t intf;          /* L3 interface */
    uint32 multicast_flags; /* BCM_L3_MULTICAST flag definitions */
} bcm_niv_egress_t;
```

OPERATIONS, ADMINISTRATION, AND MAINTENANCE

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

Table 46: OAM Loss Measurement Flag Definitions

Flag	Description
BCM_OAM_LOSS_SLM	Transmit SLM/Rs (as opposed to LMM/Rs)

A new OAM control format has been added with its data type.

Table 47: OAM Control Formats

Control Format Type	Description
bcmOamControlReportMode	Accepts one of the bcm_oam_report_mode_type_t as the arg

```
/*
 * OAM report mode types. To be used as the arg parameter when calling
 * bcm_oam_control_set/get() with the type parameter set to
 * bcmOamControlReportMode.
 */
typedef enum bcm_oam_report_mode_type_e {
    bcmOamReportModeTypeNormal = 0,   /* Default report mode. */
    bcmOamReportModeTypeCompact = 1,
    bcmOamReportModeTypeRaw = 2,      /* Report mode events will be disabled
                      * when report mode is set to raw. */
    bcmOamReportModeTypeCount = 3
} bcm_oam_report_mode_type_t;
```

PACKET TRANSMIT AND RECEIVE

Some packet stacking forwarding options have been added as listed below for egress device operations which may be specified in a HiGig2 header.

Table 48: Packet Stacking Forward Descriptions

Name	Description
BCM_PKT_STK_FORWARD_MPLS	Stacking header packet forwarding option:MPLS.

Table 48: Packet Stacking Forward Descriptions

Name	Description
BCM_PKT_STK_FORWARD_TRILL	Stacking header packet forwarding option:TRILL.
BCM_PKT_STK_FORWARD_FCOE	Stacking header packet forwarding option:FCoE.
BCM_PKT_STK_FORWARD_SNOOP	Stacking header packet forwarding option:SNOOP.
BCM_PKT_STK_FORWARD_TRAFFIC_MANAGEMENT	Stacking header packet forwarding option:Traffic Management.

Following are the new Rx Decap Tunnel types added in this release.

Table 49: Rx Decap Tunnel Types

RX Decap Tunnel Type	Description
bcmRxDecapNone	No tunnel Decap
bcmRxDecapAccessSVP	Packet ingress on Access SVP (No decap)
bcmRxDecapMIM	Decap MIM Tunnel
bcmRxDecapL2GRE	Decap L2GRE Tunnel
bcmRxDecapVXLAN	Decap VXLAN Tunnel
bcmRxDecapAMT	Decap AMT Tunnel
bcmRxDecapIP	Decap IP Tunnel
bcmRxDecapTRILL	Decap TRILL Tunnel
bcmRxDecapL2MPLS1LABEL	Decap MPLS 1 Label, L2 payload, no Control Word present
bcmRxDecapL2MPLS2LABEL	Decap MPLS 2 Label, L2 payload, no Control Word present
bcmRxDecapL2MPLS1LABELCW	Decap MPLS 1 Label, L2 payload, Control Word present
bcmRxDecapL2MPLS2LABELCW	Decap MPLS 2 Label, L2 payload, Control Word present
bcmRxDecapL3MPLS1LABEL	Decap MPLS 1 Label, L3 payload, no Control Word present
bcmRxDecapL3MPLS2LABEL	Decap MPLS 2 Label, L3 payload, no Control Word present
bcmRxDecapL3MPLS1LABELCW	Decap MPLS 1 Label, L3 payload, Control Word present
bcmRxDecapL3MPLS2LABELCW	Decap MPLS 2 Label, L3 payload, Control Word present
bcmRxDecapWTP2AC	Decap WTP2AC Tunnel
bcmRxDecapAC2AC	Decap AC2AC Tunnel

The following new elements of bcm_pkt_t structure have been added in this release with the new packet flags in flags2 field.

Table 50: bcm_pkt_t Structure Description

Field	Type	Description
rx_decap_tunnel	bcm_rx_decap_tunnel_t	For receive only, type of outer tunnel decapped. See Rx Decap Tunnel Types (page 77)
src_vport	bcm_gport_t	PPH.In-LIF-or-In-RIF field
dst_vport	bcm_gport_t	FTMH.OUTLIF_MCID field.If dest_port != -1 then outlif, otherwise MCID
fwd_hdr_offset	uint32	the Forwarding-Header-Offset (in bytes) from start of packet, i.e. PPH.FWD_HEADER_OFFSET
snoop_cmnd	int	indicates the two msb bits in Snoop-Command (PPH.SNOOP_CPU_CODE)
stk_dst_gport	bcm_gport_t	Stacking destination port (FMTH.DSP-Ext).
stk_route_tm_domains	uint32	Bitmap of the traversed TM domains (FMTH.Stacking_Route-History)

Table 50: bcm_pkt_t Structure Description

Field	Type	Description
oam_hdr_offset	uint32	Distance (in bytes) to OAM header from start of packet

Table 51: Packet flags2 Flags Descriptions

Flag	Description
BCM_PKT_F2_SNOOPED	This flag indicates the Packet was snooped.
BCM_PKT_F2_UNKNOWN_DEST	packet has unknown destination.

POLICER CONFIGURATION

New policer group modes of bcmPolicerGroupModeIntPriCascade and bcmPolicerGroupModeIntPriCascadeWithCoupling with the policer flags BCM_POLICER_BW_SHARING_GROUP_START and BCM_POLICER_BW_SHARING_GROUP_END have been added in this release.

Table 52: Policer Group Modes

Mode	Description
bcmPolicerGroupModeIntPriCascade	A set of policers(max 8) selected based on internal priority, wherein excess bandwidth can overflow from high priority bucket to low priority bucket
bcmPolicerGroupModeIntPriCascadeWithCoupling	A set of policers(max 4) selected based on internal priority, wherein excess bandwidth can overflow from high priority bucket to low priority bucket, and from low priority green bucket to high priority yellow

Table 53: Policer Flags

Name	Purpose
BCM_POLICER_BW_SHARING_GROUP_START	Policer corresponds to the start of bandwidth sharing group(cascade chain).
BCM_POLICER_BW_SHARING_GROUP_END	Policer corresponds to the end of bandwidth sharing group(cascade chain).

PORT CONFIGURATION

Two PCS controls have been added as below.

Table 54: bcm_port_pcs_t

Value	Description
bcmPortPCS64b66bRsFec	64b66bRsFec
bcmPortPCS64b66bLowLatencyRsFec	64b66bLowLatencyRsFec

Three PRBS Polynomial types have been added as below in this release.

Table 55: PRBS Values

BCM_PORT_PRBS_POLYNOMIAL_X9_X5_1	***polynomial = x^9 + x^5 + 1***
`BCM_PORT_PRBS_POLYNOMIAL_X11_X9_1`	polynomial = x^11 + x^9 + 1
`BCM_PORT_PRBS_POLYNOMIAL_X58_X31_1`	polynomial = x^58 + x^31 + 1

Below class types have been added for `bcm_port_class_set()` and `bcm_port_class_get()` functions.

Table 56: bcm_port_class_t

bcmPortClassFieldIngressVlanTranslation	***Class for field stage Ingress Vlan Translation***
bcmPortClassFieldIngressTunnelTerminated	Class for field stage Ingress Tunnel Terminated

Below port control types has been added for `bcm_port_control_set()` and `bcm_port_control_get()` functions.

Table 57: bcm_port_control_t

bcmPortControlBridge	***Enable/Disable local port bridging***
bcmPortControlLlfcCellsCongestionIndEnable	EnableDisable extracting cells congestion indication from LLFC cells

Several `BCM_PORT_PHY_CONTROL_xxx` configuration types have been added.

Table 58: bcm_port_phy_control_t

BCM_PORT_PHY_CONTROL_DIGITAL_TEMP	***Returns the internally derived temperature (celcius degree) of the die***
`BCM_PORT_PHY_CONTROL_ANALOG_TEMP`	Returns the internally derived temperature (celcius degree) of the die for the analog section
`BCM_PORT_PHY_CONTROL_TX_FIR_PRE`	
`BCM_PORT_PHY_CONTROL_TX_FIR_MAIN`	
`BCM_PORT_PHY_CONTROL_TX_FIR_POST`	
`BCM_PORT_PHY_CONTROL_TX_FIR_POST2`	
`BCM_PORT_PHY_CONTROL_TX_FIR_POST3`	
`BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_PMA_OS`	
`BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_SCR_MODE`	
`BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_ENCODE_MODE`	
`BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_CL48_CHECK_END`	
`BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_BLK_SYNC_MODE`	
`BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_REORDER_MODE`	

Table 58: bcm_port_phy_control_t

BCM_PORT_PHY_CONTROL_DIGITAL_TEMP	***Returns the internally derived temperature (celcius degree) of the die***
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_CL36_EN	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_DESCR1_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_DEC1_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_DESKEW_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_DESC2_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_DESC2_BYTE_DEL	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_BRCM64B66_DESCR	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_SGMII_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_CLKCNT0	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_CLKCNT1	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_LPCNT0	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_LPCNT1	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_MAC_CGC	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_PCS_REPCNT	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_PCS_CRDTEN	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_PCS_CLKCNT	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_PCS_CGC	
BCM_PORT_PHY_CONTROL_PCS_SPEED_HTO_CL72_EN	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_ENTRY	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_HCD	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_PMA_OS	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_SCR_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_ENCODE_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_CL48_CHECK_END	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_BLK_SYNC_MODE	

Table 58: bcm_port_phy_control_t

BCM_PORT_PHY_CONTROL_DIGITAL_TEMP	***Returns the internally derived temperature (celcius degree) of the die***
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_REORDER_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_CL36_EN	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_DESCR1_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_DEC1_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_DESKEW_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_DESC2_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_DESC2_BYTE_DEL	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_BRCM64B66_DESCR	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_SGMII_MODE	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_CLKCNT0	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_CLKCNT1	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_LPCNT0	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_LPCNT1	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_MAC_CGC	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_PCS_REPCNT	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_PCS_CRDTEN	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_PCS_CLKCNT	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_PCS_CGC	
BCM_PORT_PHY_CONTROL_PCS_SPEED_ST_CL72_EN	
BCM_PORT_PHY_CONTROL_LOOPBACK_PMD	
BCM_PORT_PHY_CONTROL_TX_PATTERN_GEN_ENABLE	
BCM_PORT_PHY_CONTROL_TX_PATTERN_LENGTH	

Two Internal Port flags as below have been added for `bcm_port_internal_get()` function.

Table 59: Internal Port Flags

Flag	*Meaning*
`BCM_PORT_INTERNAL_CONF_SCOPE_CORE0`	core 0 indication
`BCM_PORT_INTERNAL_CONF_SCOPE_CORE1`	core 1 indication

The flag that allows remapping an encapsulation to a port on a different core has been added for API `bcm_port_encap_map_set()` and `bcm_port_encap_map_get()`.

bcm_port_encap_map_set

Set the encapsulation to port mapping from encap_id to port.

Syntax

```
#include <bcm/port.h>
int
bcm_port_encap_map_set(
    int unit,
    uint32 flags,
    bcm_if_t encap_id,
    bcm_gport_t port);
```

Parameters

`unit`	(IN) Unit number
`flags`	(IN) API flags, Supported flags: BCM_PORT_ENCAP_MAP_ALLOW_CORE_CHANGE
`encap_id`	(IN) encapsulation ID to map from
`port`	(IN) port to map to

Description

Set the encapsulation to port mapping from `encap_id` to port. Currently used only for egress multicast mapping of outlif to local port. By default the API will not allow remapping an encapsulation to a port on a different core, due to hardware limitations. If using the `BCM_PORT_ENCAP_MAP_ALLOW_CORE_CHANGE` flag this will be allowed, and it would be the user's responsibility to make sure this outlif is not used in any outlif-only replication of any (egress) multicast group. Not adhering will result in wrong behavior by the hardware.

Returns

`BCM_E_XXX`

bcm_port_encap_map_get

Get the encapsulation to port mapping from encap_id.

Syntax

```
#include <bcm/port.h>
int
bcm_port_encap_map_get(
    int unit,
    uint32 flags,
    bcm_if_t encap_id,
    bcm_gport_t *port);
```

Parameters

`unit`	(IN) Unit number
`flags`	(IN) API flags, currently unused
`encap_id`	(IN) encapsulation ID mapped from
`port`	(OUT) port mapped to

Description

Get the encapsulation to port mapping from `encap_id.` Currently used only for egress multicast mapping of outlif to local port.

Returns

`BCM_E_XXX`

New `bcm_port_stat_sync_get/get32()` and `bcm_port_stat_counter_sync_get()` are added in this release.

bcm_port_stat_sync_get
bcm_port_stat_sync_get32

Force an immediate counter update and retrieve port counter value for
specified port statistic type.

Syntax

```
#include <bcm/port.h>
int
bcm_port_stat_sync_get(
    int unit,
    bcm_gport_t port,
    bcm_port_stat_t stat,
    uint64 *val);

int
bcm_port_stat_sync_get32(
    int unit,
    bcm_gport_t port,
    bcm_port_stat_t stat,
    uint32 *val);
```

Parameters

unit	(IN) Unit number.
port	(IN) GPORT ID
stat	(IN) Type of the counter to retrieve.
val	(OUT) Pointer to a counter value.

Description

Similar to `bcm_port_stat_get()`, value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_XXX
```

bcm_port_stat_counter_sync_get

Force an immediate counter update and retrieve counter values for
specified gport based on port statistic type.

Syntax

```
#include <bcm/port.h>
int
bcm_port_stat_counter_sync_get(
    int                unit,
    bcm_gport_t        port,
    bcm_port_stat_t    stat,
    uint32             num_entries,
    uint32             *counter_indexes,
    bcm_stat_value_t   *counter_values);
```

Parameters

unit	(IN) Unit number.
port	(IN) GPORT ID
stat	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
num_entries	(IN) Number of counter Entries
counter_indexes	(IN) Pointer to Counter indexes entries
counter_values	(OUT) Pointer to counter values

Description

Similar to `bcm_port_stat_counter_get()`, value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_xxx
```

PRECISION TIME PROTOCOL

The resolution of the Bridge Time is corrected to seconds. It was nanoseconds as show in API guide.

```
/* PTP Servo Configuration Data Type */
typedef struct bcm_ptp_servo_config_s {
    ...
    uint32 bridge_time;                  /* Bridge time in seconds */
    ...
} bcm_ptp_servo_config_t;
```

Three new fields of data type `bcm_ptp_foreign_master_entry_t` have been added in this release.

```
typedef struct bcm_ptp_foreign_master_entry_s {
    ...
    uint16 offset_scaled_log_variance;      /* log variance field of announce
msg. */
    bcm_ptp_clock_accuracy_t clock_accuracy; /* Clock Accuracy */
    uint16 steps_removed;                  /* steps removed field of announce
msg. */
} bcm_ptp_foreign_master_entry_t;
```

New APIs have been added as following with the required data types and defines:

```
/* ESMC definitions and data types. */
typedef enum bcm_esmc_network_option_e {
    bcm_esmc_network_option_g781_I = 1,
    bcm_esmc_network_option_g781_II = 2,
    bcm_esmc_network_option_g781_III = 3
} bcm_esmc_network_option_t;

typedef enum bcm_esmc_quality_level_e {
    bcm_esmc_g781_I_ql_prc = 0x0100,
    bcm_esmc_g781_I_ql_ssua = 0x0101,
    bcm_esmc_g781_I_ql_ssub = 0x0102,
    bcm_esmc_g781_I_ql_sec = 0x0103,
    bcm_esmc_g781_I_ql_dnu = 0x0104,
    bcm_esmc_g781_I_ql_inv0 = 0x010f,
    bcm_esmc_g781_I_ql_inv1 = 0x011f,
    bcm_esmc_g781_I_ql_inv3 = 0x013f,
    bcm_esmc_g781_I_ql_inv5 = 0x015f,
    bcm_esmc_g781_I_ql_inv6 = 0x016f,
    bcm_esmc_g781_I_ql_inv7 = 0x017f,
    bcm_esmc_g781_I_ql_inv9 = 0x019f,
    bcm_esmc_g781_I_ql_inv10 = 0x01af,
    bcm_esmc_g781_I_ql_inv12 = 0x01cf,
    bcm_esmc_g781_I_ql_inv13 = 0x01df,
    bcm_esmc_g781_I_ql_inv14 = 0x01ef,
    bcm_esmc_g781_I_ql_failed = 0x01ff,
    bcm_esmc_g781_I_ql_nsupp = 0x01ff,
    bcm_esmc_g781_I_ql_unc = 0x01ff,
    bcm_esmc_g781_II_ql_prs = 0x0200,
    bcm_esmc_g781_II_ql_stu = 0x0201,
    bcm_esmc_g781_II_ql_st2 = 0x0202,
    bcm_esmc_g781_II_ql_tnc = 0x0203,
    bcm_esmc_g781_II_ql_st3e = 0x0204,
    bcm_esmc_g781_II_ql_st3 = 0x0205,
    bcm_esmc_g781_II_ql_smc = 0x0206,
    bcm_esmc_g781_II_ql_prov = 0x0207,
    bcm_esmc_g781_II_ql_dus = 0x0208,
    bcm_esmc_g781_II_ql_inv2 = 0x022f,
    bcm_esmc_g781_II_ql_inv3 = 0x023f,
    bcm_esmc_g781_II_ql_inv5 = 0x025f,
    bcm_esmc_g781_II_ql_inv6 = 0x026f,
    bcm_esmc_g781_II_ql_inv8 = 0x028f,
    bcm_esmc_g781_II_ql_inv9 = 0x029f,
    bcm_esmc_g781_II_ql_inv11 = 0x02bf,
    bcm_esmc_g781_II_ql_failed = 0x02ff,
    bcm_esmc_g781_II_ql_nsupp = 0x02ff,
    bcm_esmc_g781_II_ql_unc = 0x02ff,
    bcm_esmc_g781_III_ql_unk = 0x0300,
    bcm_esmc_g781_III_ql_sec = 0x0301,
    bcm_esmc_g781_III_ql_inv1 = 0x031f,
    bcm_esmc_g781_III_ql_inv2 = 0x032f,
    bcm_esmc_g781_III_ql_inv3 = 0x033f,
    bcm_esmc_g781_III_ql_inv4 = 0x034f,
    bcm_esmc_g781_III_ql_inv5 = 0x035f,
    bcm_esmc_g781_III_ql_inv6 = 0x036f,
    bcm_esmc_g781_III_ql_inv7 = 0x037f,
    bcm_esmc_g781_III_ql_inv8 = 0x038f,
    bcm_esmc_g781_III_ql_inv9 = 0x039f,
    bcm_esmc_g781_III_ql_inv10 = 0x03af,
    bcm_esmc_g781_III_ql_inv12 = 0x03cf,
    bcm_esmc_g781_III_ql_inv13 = 0x03df,
    bcm_esmc_g781_III_ql_inv14 = 0x03ef,
    bcm_esmc_g781_III_ql_inv15 = 0x03ff,
    bcm_esmc_g781_III_ql_failed = 0x03ff,
    bcm_esmc_g781_III_ql_nsupp = 0x03ff,
    bcm_esmc_g781_III_ql_unc = 0x03ff,
    bcm_esmc_ql_unresolvable = 0xffff
} bcm_esmc_quality_level_t;

typedef enum bcm_esmc_pdu_type_e {
    bcm_esmc_pdu_type_info,
    bcm_esmc_pdu_type_event
} bcm_esmc_pdu_type_t;

typedef struct bcm_esmc_pdu_data_s {
    bcm_mac_t source_mac;
    bcm_esmc_pdu_type_t pdu_type;
    bcm_esmc_quality_level_t ql;
    uint8 ssm_code;
} bcm_esmc_pdu_data_t;

typedef int (*bcm_esmc_rx_cb)(
    int unit,
    int stack_id,
    int ingress_port,
    bcm_esmc_pdu_data_t *pdu_data);

/* T-DPLL definitions and data types. */
#define BCM_TDPLL_INPUT_CLOCK_NUM_GPIO  (6)
#define BCM_TDPLL_INPUT_CLOCK_NUM_SYNCE (2)
#define BCM_TDPLL_INPUT_CLOCK_NUM_1588  (2)
#define BCM_TDPLL_INPUT_CLOCK_NUM_MAX (BCM_TDPLL_INPUT_CLOCK_NUM_GPIO + \n
BCM_TDPLL_INPUT_CLOCK_NUM_SYNCE + \n           BCM_TDPLL_INPUT_CLOCK_NUM_1588)

#define BCM_TDPLL_OUTPUT_CLOCK_NUM_BROADSYNC (2)
#define BCM_TDPLL_OUTPUT_CLOCK_NUM_SYNCE     (2)
#define BCM_TDPLL_OUTPUT_CLOCK_NUM_1588      (1)
#define BCM_TDPLL_OUTPUT_CLOCK_NUM_GPIO      (6)
#define BCM_TDPLL_OUTPUT_CLOCK_NUM_MAX
(BCM_TDPLL_OUTPUT_CLOCK_NUM_BROADSYNC + \n
BCM_TDPLL_OUTPUT_CLOCK_NUM_SYNCE + \n
BCM_TDPLL_OUTPUT_CLOCK_NUM_1588 + \n
BCM_TDPLL_OUTPUT_CLOCK_NUM_GPIO)

#define BCM_TDPLL_DPLL_INSTANCE_NUM_MAX (BCM_TDPLL_OUTPUT_CLOCK_NUM_MAX - \n
BCM_TDPLL_OUTPUT_CLOCK_NUM_GPIO)

typedef enum bcm_tdpll_dpll_bandwidth_units_e {
    bcm_tdpll_dpll_bandwidth_mHz,
    bcm_tdpll_dpll_bandwidth_Hz,
    bcm_tdpll_dpll_bandwidth_kHz
} bcm_tdpll_dpll_bandwidth_units_t;

typedef struct bcm_tdpll_dpll_bandwidth_s {
    uint32 value;
    bcm_tdpll_dpll_bandwidth_units_t units;
} bcm_tdpll_dpll_bandwidth_t;

typedef enum bcm_tdpll_dpll_phase_mode_e {
    bcm_tdpll_dpll_phase_mode_none,
    bcm_tdpll_dpll_phase_mode_pbo,
    bcm_tdpll_dpll_phase_mode_pboslew
} bcm_tdpll_dpll_phase_mode_t;

typedef struct bcm_tdpll_dpll_phase_control_s {
    bcm_tdpll_dpll_phase_mode_t mode;
    int offset_ns;
} bcm_tdpll_dpll_phase_control_t;

typedef struct bcm_tdpll_dpll_properties_s {
    bcm_tdpll_dpll_bandwidth_t bandwidth;
    bcm_tdpll_dpll_phase_control_t phase_control;
} bcm_tdpll_dpll_properties_t;

typedef struct bcm_tdpll_dpll_bindings_s {
    SHR_BITDCL input_clocks[_SHR_BITDCLSIZE(BCM_TDPLL_INPUT_CLOCK_NUM_MAX)];
```

```
   SHR_BITDCL output_clocks[_SHR_BITDCLSIZE(BCM_TDPLL_OUTPUT_CLOCK_NUM_MAX)];
} bcm_tdpll_dpll_bindings_t;

typedef struct bcm_tdpll_dpll_instance_s {
    int index;
    bcm_tdpll_dpll_bindings_t bindings;
    bcm_tdpll_dpll_properties_t properties;
    int reference;
} bcm_tdpll_dpll_instance_t;

typedef struct bcm_tdpll_input_clock_l1mux_s {
    int index;
    int port;
} bcm_tdpll_input_clock_l1mux_t;

typedef struct bcm_tdpll_input_clock_frequency_s {
    uint32 clock;
    uint32 tsevent;
    int tsevent_quotient;
} bcm_tdpll_input_clock_frequency_t;

typedef enum bcm_tdpll_input_clock_monitor_type_e {
    bcm_tdpll_input_clock_monitor_type_soft_warn,
    bcm_tdpll_input_clock_monitor_type_hard_accept,
    bcm_tdpll_input_clock_monitor_type_hard_reject
} bcm_tdpll_input_clock_monitor_type_t;

typedef struct bcm_tdpll_input_clock_monitor_s {
    uint64 tsevent_dt;
    uint64 tsevent_time;
    uint64 tsevent_num;
    uint64 dt_ns;
    uint64 dtref_ns;
    uint64 dt_sum_ns;
    uint64 dtref_sum_ns;
    uint64 prior_evnum;
    uint32 numev_sum;
    int freq_error_ppb;
    int over_soft_warn_threshold;
    int under_hard_accept_threshold;
    int over_hard_reject_threshold;
    uint32 num_missing_tsevent;
} bcm_tdpll_input_clock_monitor_t;

typedef struct bcm_tdpll_input_clock_select_s {
    bcm_esmc_quality_level_t ql;
    int priority;
    int lockout;
} bcm_tdpll_input_clock_select_t;

typedef struct bcm_tdpll_input_clock_s {
    int index;
    bcm_tdpll_input_clock_l1mux_t l1mux;
    bcm_mac_t mac;
    uint32 state;
    bcm_tdpll_input_clock_frequency_t frequency;
    bcm_tdpll_input_clock_monitor_t monitor;
    bcm_tdpll_input_clock_select_t select;
    int dpll_use[BCM_TDPLL_DPLL_INSTANCE_NUM_MAX];
} bcm_tdpll_input_clock_t;

typedef struct bcm_tdpll_input_clock_monitor_cb_data_s {
    int index;
    bcm_tdpll_input_clock_monitor_type_t monitor_type;
    int monitor_value;
} bcm_tdpll_input_clock_monitor_cb_data_t;

typedef int (*bcm_tdpll_input_clock_monitor_cb)(
    int unit,
    int stack_id,
    bcm_tdpll_input_clock_monitor_cb_data_t *cb_data);

typedef struct bcm_tdpll_input_clock_selector_cb_data_s {
    int dpll_index;
    int prior_selected_clock;
    int selected_clock;
} bcm_tdpll_input_clock_selector_cb_data_t;

typedef int (*bcm_tdpll_input_clock_selector_cb)(
    int unit,
    int stack_id,
    bcm_tdpll_input_clock_selector_cb_data_t *cb_data);

typedef struct bcm_tdpll_output_clock_frequency_s {
    uint32 synth;
    uint32 tsevent;
    int tsevent_quotient;
    uint32 deriv;
    int deriv_quotient;
} bcm_tdpll_output_clock_frequency_t;

typedef struct bcm_tdpll_output_clock_s {
    int index;
    uint32 state;
    bcm_tdpll_output_clock_frequency_t frequency;
} bcm_tdpll_output_clock_t;

typedef enum bcm_tdpll_holdover_mode_e {
    bcm_tdpll_holdover_mode_instantaneous,
    bcm_tdpll_holdover_mode_avg1s,
    bcm_tdpll_holdover_mode_manual,
    bcm_tdpll_holdover_mode_fast_average,
    bcm_tdpll_holdover_mode_slow_average
} bcm_tdpll_holdover_mode_t;

/* Frequency correction */
```

```
typedef int32 bcm_tdpll_frequency_correction_t;

typedef struct bcm_tdpll_holdover_data_s {
    bcm_tdpll_frequency_correction_t freq_instantaneous;
    bcm_tdpll_frequency_correction_t freq_avg1s;
    bcm_tdpll_frequency_correction_t freq_manual;
    bcm_tdpll_frequency_correction_t freq_fast_average;
    bcm_tdpll_frequency_correction_t freq_slow_average;
    int freq_fast_average_valid;
    int freq_slow_average_valid;
    bcm_tdpll_holdover_mode_t mode;
} bcm_tdpll_holdover_data_t;
```

bcm_esmc_tx

ESMC PDU transmit (Tx).

Syntax

```
#include <bcm/ptp.h>
int
bcm_esmc_tx(
    int unit,
    int stack_id,
    bcm_pbmp_t pbmp,
    bcm_esmc_pdu_data_t *esmc_pdu_data);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`pbmp`	(IN) Tx port bitmap.
`esmc_pdu_data(IN)`	ESMC PDU data.

Description

ESMC PDU transmit (Tx).

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_esmc_rx_callback_register

Register ESMC PDU Rx callback.

Syntax

```
#include <bcm/ptp.h>
int
bcm_esmc_rx_callback_register(
    int unit,
    int stack_id,
    bcm_esmc_rx_cb rx_cb);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
rx_cb	(IN) ESMC PDU Rx callback function pointer.

Description

Register ESMC PDU Rx callback.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_esmc_rx_callback_unregister

Unregister ESMC PDU Rx callback.

Syntax

```
#include <bcm/ptp.h>
int
bcm_esmc_rx_callback_unregister(
    int unit,
    int stack_id);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.

Description

Unregister ESMC PDU Rx callback.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_esmc_tunnel_get

Get ESMC PDU tunneling-enabled Boolean.

Syntax

```
#include <bcm/ptp.h>
int
bcm_esmc_tunnel_get(
    int unit,
    int stack_id,
    int *enable);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`enable`	(OUT) Enable Boolean.

Description

Get ESMC PDU tunneling-enabled Boolean.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_esmc_tunnel_set

Set ESMC PDU tunneling-enabled Boolean.

Syntax

```
#include <bcm/ptp.h>
int
bcm_esmc_tunnel_set(
    int unit,
    int stack_id,
    int enable);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`enable`	(IN) Enable Boolean.

Description

Set ESMC PDU tunneling-enabled Boolean.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_esmc_g781_option_get

Get ITU-T G.781 networking option for SyncE.

Syntax

```
#include <bcm/ptp.h>
int
bcm_esmc_g781_option_get(
    int unit,
    int stack_id,
    bcm_esmc_network_option_t *g781_option);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`g781_option`	(OUT) ITU-T G.781 networking option.

Description

Get ITU-T G.781 networking option for SyncE.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_esmc_g781_option_set

Set ITU-T G.781 networking option for SyncE.

Syntax

```
#include <bcm/ptp.h>
int
bcm_esmc_g781_option_set(
    int unit,
    int stack_id,
    bcm_esmc_network_option_t g781_option);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`g781_option`	(IN) ITU-T G.781 networking option.

Description

Set ITU-T G.781 networking option for SyncE.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_esmc_QL_SSM_map

Get synchronization status message (SSM) code corresponding to ITU-T G.781 quality level (QL).

Syntax

```
#include <bcm/ptp.h>
int
bcm_esmc_QL_SSM_map(
    int unit,
    bcm_esmc_network_option_t opt,
    bcm_esmc_quality_level_t ql,
    uint8 *ssm_code);
```

Parameters

`unit`	(IN) Unit number.
`opt`	(IN) ITU-T G.781 networking option.
`ql`	(IN) ITU-T G.781 quality level.
`ssm_code`	(OUT) SSM code.

Description

Get synchronization status message (SSM) code corresponding to ITU-T G.781 quality level (QL).

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_esmc_SSM_QL_map

Get ITU-T G.781 quality level (QL) corresponding to synchronization
status message (SSM) code.

Syntax

```
#include <bcm/ptp.h>
int
bcm_esmc_SSM_QL_map(
    int unit,
    bcm_esmc_network_option_t opt,
    uint8 ssm_code,
    bcm_esmc_quality_level_t *ql);
```

Parameters

`unit`	(IN) Unit number.
`opt`	(IN) ITU-T G.781 networking option.
`ssm_code`	(IN) SSM code.
`ql`	(OUT) ITU-T G.781 quality level.

Description

Get ITU-T G.781 quality level (QL) corresponding to synchronization status message (SSM) code.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_dpll_bindings_get

Get logical DPLL instance input/output bindings.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_dpll_bindings_get(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_tdpll_dpll_bindings_t *bindings);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`bindings`	(OUT) DPLL instance bindings.

Description

Get logical DPLL instance input/output bindings.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_dpll_bindings_set

Set logical DPLL instance input/output bindings.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_dpll_bindings_set(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_tdpll_dpll_bindings_t *bindings);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`bindings`	(IN) DPLL instance bindings.

Description

Set logical DPLL instance input/output bindings.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_dpll_reference_get

Get reference clocks for DPLL instances.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_dpll_reference_get(
    int unit,
    int stack_id,
    int max_num_dpll,
    int *dpll_ref,
    int *num_dpll);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`max_num_dpll`	(IN) Maximum number of DPLL instances.
`dpll_ref`	(OUT) DPLL instance reference clocks.
`num_dpll`	(OUT) Number of DPLL instances.

Description

Get reference clocks for DPLL instances.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_dpll_bandwidth_get

Get DPLL bandwidth.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_dpll_bandwidth_get(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_tdpll_dpll_bandwidth_t *bandwidth);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`bandwidth`	(OUT) DPLL bandwidth.

Description

Get DPLL bandwidth.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_dpll_bandwidth_set

Set DPLL bandwidth.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_dpll_bandwidth_set(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_tdpll_dpll_bandwidth_t *bandwidth);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`bandwidth`	(IN) DPLL bandwidth.

Description

Set DPLL bandwidth.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_dpll_phase_control_get

Get DPLL instance's phase control configuration.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_dpll_phase_control_get(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_tdpll_dpll_phase_control_t *phase_control);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`phase_control(OUT)`	Phase control configuration.

Description

Get DPLL instance's phase control configuration.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_dpll_phase_control_set

Set DPLL instance's phase control configuration.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_dpll_phase_control_set(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_tdpll_dpll_phase_control_t *phase_control);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`phase_control(IN)`	Phase control configuration.

Description

Set DPLL instance's phase control configuration.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_control

Start/stop T-DPLL input clock monitoring, reference selection, and switching state machine

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_control(
    int unit,
    int stack_id,
    int enable);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
enable	(IN) Enable Boolean.

Description

Start/stop T-DPLL input clock monitoring, reference selection, and switching state machine

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_input_clock_mac_get

Get MAC address of input clock.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_mac_get(
    int unit,
    int stack_id,
    int clock_index,
    bcm_mac_t *mac);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
clock_index	(IN) Input clock index.
mac	(OUT) Input clock MAC address.

Description

Get MAC address of input clock.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_input_clock_mac_set

Set MAC address of input clock.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_mac_set(
    int unit,
    int stack_id,
    int clock_index,
    bcm_mac_t *mac);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
clock_index	(IN) Input clock index.
mac	(IN) Input clock MAC address.

Description

Set MAC address of input clock.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_input_clock_frequency_error_get

Get fractional frequency error of an input clock from input-clock monitoring process.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_frequency_error_get(
    int unit,
    int stack_id,
    int clock_index,
    int *freq_error_ppb);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
clock_index	(IN) Input clock index.
freq_error_ppb(OUT)	Input clock fractional frequency error (ppb).

Description

Get fractional frequency error of an input clock from input-clock monitoring process.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_input_clock_threshold_state_get

Get monitor threshold state of an input clock from input-clock monitoring process.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_threshold_state_get(
    int unit,
    int stack_id,
    int clock_index,
    bcm_tdpll_input_clock_monitor_type_t threshold_type,
    int *threshold_state);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
clock_index	(IN) Input clock index.
threshold_type(IN)	Input-clock monitoring threshold type.
threshold_state(OUT)	Input-clock monitoring threshold state Boolean.

Description

Get monitor threshold state of an input clock from input-clock monitoring process.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_enable_get

Get input clock enable Boolean.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_enable_get(
    int unit,
    int stack_id,
    int clock_index,
    int *enable);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`enable`	(OUT) Input clock enable Boolean.

Description

Get input clock enable Boolean.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_enable_set

Set input-clock enable Boolean.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_enable_set(
    int unit,
    int stack_id,
    int clock_index,
    int enable);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
clock_index	(IN) Input clock index.
enable	(IN) Input clock enable Boolean.

Description

Set input-clock enable Boolean.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_input_clock_l1mux_get

Get L1 mux mapping (mux index and port number) of input clock.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_l1mux_get(
    int unit,
    int stack_id,
    int clock_index,
    bcm_tdpll_input_clock_l1mux_t *l1mux);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
clock_index	(IN) Input clock index.
l1mux	(OUT) L1 mux mapping.

Description

Get L1 mux mapping (mux index and port number) of input clock.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_l1mux_set

Set L1 mux mapping (mux index and port number) of input clock.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_l1mux_set(
    int unit,
    int stack_id,
    int clock_index,
    bcm_tdpll_input_clock_l1mux_t *l1mux);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`l1mux`	(IN) L1 mux mapping.

Description

Set L1 mux mapping (mux index and port number) of input clock.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_valid_get

Get valid Boolean of an input clock from input-clock monitoring process.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_valid_get(
    int unit,
    int stack_id,
    int clock_index,
    int *valid);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`valid`	(OUT) Input clock valid Boolean.

Description

Get valid Boolean of an input clock from input-clock monitoring process.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_valid_set

Set input-clock valid Boolean from monitoring process.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_valid_set(
    int unit,
    int stack_id,
    int clock_index,
    int valid);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`valid`	(IN) Input clock valid Boolean.

Description

Set input-clock valid Boolean from monitoring process.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_frequency_get

Get input clock frequency.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_frequency_get(
    int unit,
    int stack_id,
    int clock_index,
    uint32 *clock_frequency,
    uint32 *tsevent_frequency);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`clock_frequency(OUT)`	Frequency (Hz).
`tsevent_frequency(OUT)`	TS event frequency (Hz).

Description

Get input clock frequency.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_frequency_set

Set input clock frequency.

Syntax

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

```
int
bcm_tdpll_input_clock_frequency_set(
    int unit,
    int stack_id,
    int clock_index,
    uint32 clock_frequency,
    uint32 tsevent_frequency);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`clock_frequency(IN)`	Frequency (Hz).
`tsevent_frequency(IN)`	TS event frequency (Hz).

Description

Set input clock frequency.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_ql_get

Get input clock quality level (QL).

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_ql_get(
    int unit,
    int stack_id,
    int clock_index,
    bcm_esmc_quality_level_t *ql);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`ql`	(OUT) QL.

Description

Get input clock quality level (QL).

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_ql_set

Set input clock quality level (QL).

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_ql_set(
    int unit,
    int stack_id,
    int clock_index,
    bcm_esmc_quality_level_t ql);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`ql`	(IN) QL.

Description

Set input clock quality level (QL).

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_priority_get

Get input clock priority for reference selection.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_priority_get(
    int unit,
    int stack_id,
    int clock_index,
    int *priority);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`priority`	(OUT) Input clock priority.

Description

Get input clock priority for reference selection.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_priority_set

Set input clock priority for reference selection.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_priority_set(
    int unit,
    int stack_id,
    int clock_index,
    int priority);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`priority`	(IN) Input clock priority.

Description

Set input clock priority for reference selection.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_lockout_get

Get input clock lockout Boolean for reference selection.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_lockout_get(
    int unit,
    int stack_id,
    int clock_index,
    int *lockout);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`lockout`	(OUT) Input clock lockout Boolean.

Description

Get input clock lockout Boolean for reference selection.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_lockout_set

Set input clock lockout Boolean for reference selection.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_lockout_set(
    int unit,
    int stack_id,
    int clock_index,
    int lockout);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Input clock index.
`lockout`	(IN) Input clock lockout Boolean.

Description

Set input clock lockout Boolean for reference selection.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_monitor_interval_get

Get input clock monitoring interval.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_monitor_interval_get(
    int unit,
    int stack_id,
    uint32 *monitor_interval);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`monitor_interval(OUT)`	Input clock monitoring interval (sec).

Description

Get input clock monitoring interval.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_monitor_interval_set

Set input clock monitoring interval.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_monitor_interval_set(
    int unit,
    int stack_id,
    uint32 monitor_interval);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`monitor_interval(IN)`	Input clock monitoring interval (sec).

Description

Set input clock monitoring interval.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_input_clock_monitor_threshold_get

Get monitor threshold for input-clock valid classification required in reference selection.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_monitor_threshold_get(
    int unit,
    int stack_id,
    bcm_tdpll_input_clock_monitor_type_t threshold_type,
    uint32 *threshold);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
threshold_type(IN)	Input clock monitoring threshold type.
threshold	(OUT) Input clock monitoring threshold (ppb).

Description

Get monitor threshold for input-clock valid classification required in reference selection.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_input_clock_monitor_threshold_set

Set monitor threshold for input-clock valid classification required in reference selection.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_monitor_threshold_set(
    int unit,
    int stack_id,
    bcm_tdpll_input_clock_monitor_type_t threshold_type,
    uint32 threshold);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
threshold_type(IN)	Input-clock monitoring threshold type.
threshold	(IN) Input-clock monitoring threshold (ppb).

Description

Set monitor threshold for input-clock valid classification required in reference selection.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_input_clock_ql_enabled_get

Get QL-enabled Boolean for reference selection.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_ql_enabled_get(
    int unit,
    int stack_id,
    int dpll_index,
    int *ql_enabled);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
dpll_index	(IN) DPLL instance number.
ql_enabled	(OUT) QL-enabled Boolean.

Description

Get QL-enabled Boolean for reference selection.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_ql_enabled_set

Set QL-enabled Boolean for reference selection.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_ql_enabled_set(
    int unit,
    int stack_id,
    int dpll_index,
    int ql_enabled);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`ql_enabled`	(IN) QL-enabled Boolean.

Description

Set QL-enabled Boolean for reference selection.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_revertive_get

Get revertive mode Boolean for reference selection and switching.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_revertive_get(
    int unit,
    int stack_id,
    int dpll_index,
    int *revertive);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`revertive`	(OUT) Revertive mode Boolean.

Description

Get revertive mode Boolean for reference selection and switching.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_revertive_set

Set revertive mode Boolean for reference selection and switching.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_revertive_set(
    int unit,
    int stack_id,
    int dpll_index,
    int revertive);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`revertive`	(IN) Revertive mode Boolean.

Description

Set revertive mode Boolean for reference selection and switching.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_best_get

Get best (i.e. selected) reference for a DPLL instance.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_best_get(
    int unit,
    int stack_id,
    int dpll_index,
    int *best_clock);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`best_clock`	(OUT) Best / preferred input clock index.

Description

Get best (i.e. selected) reference for a DPLL instance.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_monitor_callback_register

Register input clock monitoring callback.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_monitor_callback_register(
    int unit,
    int stack_id,
    bcm_tdpll_input_clock_monitor_cb monitor_cb);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`monitor_cb`	(IN) Input clock monitoring callback function pointer.

Description

Register input clock monitoring callback.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_monitor_callback_unregister

Unregister input clock monitoring callback.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_monitor_callback_unregister(
    int unit,
    int stack_id);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.

Description

Unregister input clock monitoring callback.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_selector_callback_register

Register input clock reference selection callback.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_selector_callback_register(
    int unit,
    int stack_id,
    bcm_tdpll_input_clock_selector_cb selector_cb);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`selector_cb`	(IN) Input clock selection callback function pointer

Description

Register input clock reference selection callback.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_input_clock_selector_callback_unregister

Unregister input clock reference selection callback.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_input_clock_selector_callback_unregister(
    int unit,
    int stack_id);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.

Description

Unregister input clock reference selection callback.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_output_clock_enable_get

Get output clock enable Boolean.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_output_clock_enable_get(
    int unit,
    int stack_id,
    int clock_index,
    int *enable);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Output clock index.
`enable`	(OUT) Output clock enable Boolean.

Description

Get output clock enable Boolean.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_output_clock_enable_set

Set output clock enable Boolean.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_output_clock_enable_set(
    int unit,
    int stack_id,
    int clock_index,
    int enable);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Output clock index.
`enable`	(IN) Output clock enable Boolean.

Description

Set output clock enable Boolean.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_output_clock_synth_frequency_get

Get output-clock (synthesizer) frequency.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_output_clock_synth_frequency_get(
    int unit,
    int stack_id,
    int clock_index,
    uint32 *synth_frequency,
    uint32 *tsevent_frequency);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Output clock index.
`synth_frequency(OUT)`	Synthesizer frequency (Hz).
`tsevent_frequency(OUT)`	TS event frequency (Hz).

Description

Get output-clock (synthesizer) frequency.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_output_clock_synth_frequency_set

Set output-clock (synthesizer) frequency.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_output_clock_synth_frequency_set(
    int unit,
    int stack_id,
    int clock_index,
    uint32 synth_frequency,
    uint32 tsevent_frequency);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Output clock index.
`synth_frequency(IN)`	Synthesizer frequency (Hz).
`tsevent_frequency(IN)`	TS event frequency (Hz).

Description

Set output-clock (synthesizer) frequency.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_output_clock_deriv_frequency_get

Get synthesizer derivative-clock frequency.

Syntax

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

```
bcm_tdpll_output_clock_deriv_frequency_get(
    int unit,
    int stack_id,
    int clock_index,
    uint32 *deriv_frequency);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Output clock index.
`deriv_frequency(OUT)`	Derivative clock frequency (Hz).

Description

Get synthesizer derivative-clock frequency.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_output_clock_deriv_frequency_set

Set synthesizer derivative clock frequency.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_output_clock_deriv_frequency_set(
    int unit,
    int stack_id,
    int clock_index,
    uint32 deriv_frequency);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Output clock index.
`deriv_frequency(OUT)`	Derivative clock frequency (Hz).

Description

Set synthesizer derivative clock frequency.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_output_clock_holdover_data_get

Get holdover configuration data.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_output_clock_holdover_data_get(
    int unit,
    int stack_id,
    int clock_index,
    bcm_tdpll_holdover_data_t *hdata);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
clock_index	(IN) Output clock index.
hdata	(OUT) Holdover configuration data.

Description

Get holdover configuration data.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_output_clock_holdover_frequency_set

Set manual holdover frequency correction.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_output_clock_holdover_frequency_set(
    int unit,
    int stack_id,
    int clock_index,
    bcm_tdpll_frequency_correction_t hfreq);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
clock_index	(IN) Output clock index.
hfreq	(IN) Holdover frequency correction (ppt).

Description

Set manual holdover frequency correction.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_output_clock_holdover_mode_get

Get holdover mode.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_output_clock_holdover_mode_get(
    int unit,
    int stack_id,
    int clock_index,
    bcm_tdpll_holdover_mode_t *hmode);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
clock_index	(IN) Output clock index.
hmode	(OUT) Holdover mode.

Description

Get holdover mode.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_output_clock_holdover_mode_set

Set holdover mode.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_output_clock_holdover_mode_set(
    int unit,
    int stack_id,
    int clock_index,
    bcm_tdpll_holdover_mode_t hmode);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`clock_index`	(IN) Output clock index
`hmode`	(IN) Holdover mode.

Description

Set holdover mode.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_output_clock_holdover_reset

Reset holdover frequency calculations.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_output_clock_holdover_reset(
    int unit,
    int stack_id,
    int clock_index);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
clock_index	(IN) Output clock index.

Description

Reset holdover frequency calculations.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_esmc_rx_state_machine

Execute state machine for Rx ESMC PDU.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_rx_state_machine(
    int unit,
    int stack_id,
    int ingress_port,
    bcm_esmc_pdu_data_t *esmc_pdu_data);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
ingress_port	(IN) Ingress port.
esmc_pdu_data(IN)	ESMC PDU.

Description

Execute state machine for Rx ESMC PDU.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_ql_get

Get quality level (QL) for ESMC.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_ql_get(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_esmc_quality_level_t *ql);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`ql`	(OUT) QL.

Description

Get quality level (QL) for ESMC.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_ql_set

Set quality level (QL) for ESMC.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_ql_set(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_esmc_quality_level_t ql);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
dpll_index	(IN) DPLL instance number.
ql	(IN) QL.

Description

Set quality level (QL) for ESMC.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_esmc_holdover_ql_get

Get quality level (QL) for ESMC during holdover.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_holdover_ql_get(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_esmc_quality_level_t *ql);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
dpll_index	(IN) DPLL instance number.
ql	(OUT) QL.

Description

Get quality level (QL) for ESMC during holdover.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_holdover_ql_set

Set quality level (QL) for ESMC during holdover.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_holdover_ql_set(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_esmc_quality_level_t ql);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`ql`	(IN) QL.

Description

Set quality level (QL) for ESMC during holdover.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_mac_get

Get MAC address for ESMC.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_mac_get(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_mac_t *mac);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
dpll_index	(IN) DPLL instance number.
mac	(OUT) Local port MAC address.

Description

Get MAC address for ESMC.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_esmc_mac_set

Set MAC address for ESMC.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_mac_set(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_mac_t *mac);
```

Parameters

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
dpll_index	(IN) DPLL instance number.
mac	(IN) Local port MAC address.

Description

Set MAC address for ESMC.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_rx_enable_get

Get ESMC PDU receive (Rx) enable state.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_rx_enable_get(
    int unit,
    int stack_id,
    int *enable);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`enable`	(OUT) ESMC PDU receive enable Boolean.

Description

Get ESMC PDU receive (Rx) enable state.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_rx_enable_set

Set ESMC PDU receive (Rx) enable state.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_rx_enable_set(
    int unit,
    int stack_id,
    int enable);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`enable`	(IN) ESMC PDU receive enable Boolean.

Description

Set ESMC PDU receive (Rx) enable state.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_tx_enable_get

Get ESMC PDU transmit (Tx) enable state.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_tx_enable_get(
    int unit,
    int stack_id,
    int dpll_index,
    int *enable);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`enable`	(OUT) ESMC PDU transmit enable Boolean.

Description

Get ESMC PDU transmit (Tx) enable state.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_tx_enable_set

Set ESMC PDU transmit (Tx) enable state.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_tx_enable_set(
    int unit,
    int stack_id,
    int dpll_index,
    int enable);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`enable`	(IN) ESMC PDU transmit enable Boolean.

Description

Set ESMC PDU transmit (Tx) enable state.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_rx_portbitmap_get

Get port bitmap for ESMC Rx.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_rx_portbitmap_get(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_pbmp_t *pbmp);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`pbmp`	(OUT) Rx port bitmap.

Description

Get port bitmap for ESMC Rx.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_rx_portbitmap_set

Set port bitmap for ESMC Rx.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_rx_portbitmap_set(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_pbmp_t pbmp);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`pbmp`	(IN) Rx port bitmap.

Description

Set port bitmap for ESMC Rx.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_tx_portbitmap_get

Get port bitmap for ESMC Tx.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_tx_portbitmap_get(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_pbmp_t *pbmp);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`pbmp`	(IN) Tx port bitmap.

Description

Get port bitmap for ESMC Tx.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

bcm_tdpll_esmc_tx_portbitmap_set

Set port bitmap for ESMC Tx.

Syntax

```
#include <bcm/ptp.h>
int
bcm_tdpll_esmc_tx_portbitmap_set(
    int unit,
    int stack_id,
    int dpll_index,
    bcm_pbmp_t pbmp);
```

Parameters

`unit`	(IN) Unit number.
`stack_id`	(IN) Stack identifier index.
`dpll_index`	(IN) DPLL instance number.
`pbmp`	(IN) Tx port bitmap.

Description

Set port bitmap for ESMC Tx.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_xxx`	Operation failed

RATE LIMITING

One new Rate Control flag has been added for the rate-limiting functions.

Table 60: Rate Control Flags

Name	*Purpose*
`BCM_RATE_COLOR_BLIND`	Rate limiting is color blind

SERVICE ACTIVATION TEST (SAT) IS NEW TO THIS RELEASE

SAT OVERVIEW

Service Activation Test (SAT) feature is basically a validation of an Ethernet Service for a customer by a service provider operator before activating the service. The customer may be another operator, upstream or downstream, or an end-point customer. It is usually done with an Ethernet Test Equipment and involves a Truck Roll to the premises where the network endpoint or the intermediate point resides. The industry is moving towards the use of embedded test application within the network switch element.

The SAT implementation borrows concept of MEP points from Ethernet OAM. Each SAT MEP has an Rx and Tx point. These SAT MEP Tx/Rx points in SAT context are defined as Service Activation Measurement Points (SAMP). SAT MEP Rx point is where a SAT packet coming from wire is trapped and sent to OLP. SAT MEP Tx point is where a SAT packet is injected from OLP before going out on wire.

A SAT MEP (SAMP) can perform one or all of the functions defined below:

- Generator Function (GTF): SAT logic that injects test frames for OVC/EVC at specified rate and Test frames processed like customer service frames.
- Collector Function (CTF): SAT Test frames are processed like customer service frames and logic has

instrumentation counters for SAT.

- Reflector Function (Latching Loopback Function): Loops back customer service frames at remote end for two-way measurement with GTF and CTF at the same end.

SAT GTF, CTF and Loopback Latch functions defined above are implemented in SAT Engine that is part of the OMAP engine. Switch that implements OAMP engine supports processing of SAT packets inside the switch.Alternatively users can interface external SAT Engines to switch as Offload Processor (OLP) connected to one of the front panel ports on the switch to process SAT packets.

To support seamless transfer of SAT packets to/from internal/external SAT engine switch implements "trap" and "inject" points to perform proxy CTF and proxy GTF functions on behalf of SAT engines. The "trap" point performs proxy CTF function and is used to receive/terminate SAT packets and redirect them to SAT engine where actual CTF function is implemented. The "inject" point performs the proxy GTF function and is used for transmitting of SAT packets generated by GTF available in SAT engine.

For devices which support SAT the APIs provide control of the following features:

- Proxy CTF Function to receive/terminate SAT packets and redirect them to OLP.
- Proxy GTF Function for initiating/transmitting SAT packets generated from OLP.
- Latching loop back Function to identify and trap all packets matching the latching loopback session to OLP/CPU with OLP header.

SAT DATA TYPES

- `bcm_sat_endpoint_t` is an identifier used to refer to an SAT endpoint.
- `bcm_sat_timestamp_format_t` is an enumeration of SAT timestamp format

The following types are used for supplying function pointers to callbacks for endpoint traversal handling.

```
typedef int (*bcm_sat_endpoint_traverse_cb)(
        int unit,
        bcm_sat_endpoint_info_t *endpoint_info,
        void *user_data);
```

SAT ENDPOINT INFO

SAT endpoint info structure. This is equivalent to an 802.1ag Maintenance Endpoint (MEP):

```
typedef struct bcm_sat_endpoint_info_s {
    bcm_sat_endpoint_t ep_id;           /* The ID associated with this
                       endpoint */
    uint32 flags;                       /* The flags associated with this
                       endpoint */
    bcm_gport_t src_gport;             /* The source gport associated with this
                       endpoint */
    bcm_vlan_t outer_vlan;              /* Outer-most VLAN tag (vlanID+ 3bits
                       dot1p +CFI) associated with this
                       endpoint */
    bcm_vlan_t inner_vlan;              /* Configure with CVLAN tag (vlanID+
                       3bits dot1p +CFI) for two-tag
                       operation or set to 0 for one-tag */
    bcm_mac_t dst_mac_address;          /* The destination MAC address
                       associated with this endpoint */
    bcm_mac_t src_mac_address;          /* The source MAC address associated
```

```
                       with this endpoint */
    uint16 ether_type;                /* Ether type of the packet */
    uint32 action_flags;              /* SAT action flags */
    uint8 pkt_pri;                    /* Egress priority marking for packet
                       redirected to DGPP */
    bcm_gport_t dest_gport;          /* Local endpoint SAT PDUs are forwarded
                       to this OLP destination port for
                       processing */
    bcm_sat_timestamp_format_t timestamp_format; /* SAT timestamp format */
    uint16 session_id;                /* SAT session ID to identify the test
                       among the MEPs */

} bcm_sat_endpoint_info_t;
```

Table 61: SAT Endpoint Structure Flag Definitions

Flag	Description
BCM_SAT_ENDPOINT_WITH_ID	Use the specified endpoint ID when creating a new endpoint
BCM_SAT_ENDPOINT_REPLACE	Replace an existing SAT endpoint
BCM_SAT_ENDPOINT_UPMEP	Endpoint associated with SAT UP-MEP
BCM_SAT_ENDPOINT_DOWNMEP	Endpoint associated with SAT DOWN-MEP
BCM_SAT_ENDPOINT_MATCH_INNER_VLAN	Selection of SAT MEP based on C VLAN.
BCM_SAT_ENDPOINT_MATCH_OUTER_VLAN	Selection of SAT MEP based on S VLAN.

Table 62: SAT Timestamp Formats

Timestamp Format Type	Description
bcmSATTimestampFormatIEEE1588v1	Low-order 64 bits of the IEEE 1588-2008 (1588v2) Precision Time Protocol timestamp format [IEEE1588]
bcmSATTimestampFormatNTP	Network Time Protocol version 4 64-bit timestamp format [RFC5905]

SAT ENDPOINT ACTION

Table 63: SAT Endpoint Action Formats

Acion Format Type	Description
BCM_SAT_ACTION_FWD_ACTION_DROP	Drop the SAT packet.
BCM_SAT_ACTION_FWD_ACTION_REDIRECT	Forward SAT packet to specified destination port.
BCM_SAT_ACTION_FWD_ACTION_LLF	Perform latching loop back function (LLF) and redirect SAT packet back to source port on which the SAT packet has been received.
BCM_SAT_ACTION_COPY_TO_CPU	Copy SAT packet to CPU.
BCM_SAT_ACTION_SAMPLE_TIMESTAMP	Sample time stamp into the OLP header.

SAT APIS

bcm_sat_endpoint_create

Create or replace an SAT endpoint object.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_endpoint_create(int unit, bcm_sat_endpoint_info_t *endpoint_info);
```

Parameters

unit	(IN) BCM device number
endpoint_info	(IN/OUT) Pointer to SAT endpoint info structure

Description

If BCM_SAT_ENDPOINT_REPLACE is not set, a new SAT endpoint is created. If BCM_SAT_ENDPOINT_WITH_ID is not set, an ID is created for the endpoint. Otherwise, the specified ID is used.

If BCM_SAT_ENDPOINT_REPLACE is set, the endpoint with the specified ID is updated with the information in the structure.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INIT	Module not initialized
BCM_E_NOT_FOUND	Attempt to replace an endpoint which does not exist
BCM_E_EXISTS	Attempt to create an endpoint with a specified ID which is already in use
BCM_E_PARAM	Attempt to replace an endpoint with no ID specified / Specified endpoint ID is out of range / An unsupported gport type was specified
BCM_E_UNAVAIL	A specified endpoint type or flag is not supported on the specified unit
BCM_E_FULL	Endpoint table is full
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

bcm_sat_endpoint_get

Get an SAT endpoint object.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_endpoint_get(int unit, bcm_sat_endpoint_t endpoint,
uint32 flags, bcm_sat_endpoint_info_t *endpoint_info);
```

Parameters

unit	(IN) BCM device number
endpoint	(IN) ID of the endpoint to get
flags	(IN) Endpoint type(UP/Down MEP) to get
endpoint_info	(OUT) Pointer to SAT endpoint info structure

Description

Retrieves the information for the endpoint with the specified ID.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INIT	Module not initialized
BCM_E_NOT_FOUND	Attempt to get an endpoint which does not exist
BCM_E_PARAM	Specified endpoint ID is out of range
BCM_E_INTERNAL	Unable to release resource lock / Failed to read memory or read or write register

bcm_sat_endpoint_destroy

Destroy an SAT endpoint object.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_endpoint_destroy(int unit, bcm_sat_endpoint_t endpoint,
     uint32 flags);
```

Parameters

unit	(IN) BCM device number
endpoint	(IN) ID of the endpoint to destroy
flags	(IN) Endpoint type (UP/Down MEP) to destroy

Description

Destroys the endpoint with the specified ID.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_TIMEOUT`	Unable to obtain resource lock
`BCM_E_INIT`	Module not initialized
`BCM_E_NOT_FOUND`	Attempt to destroy an endpoint which does not exist
`BCM_E_PARAM`	Specified endpoint ID is out of range
`BCM_E_INTERNAL`	Unable to release resource lock / Failed to read or write memory or register

bcm_sat_endpoint_destroy_all

Destroy all SAT endpoint objects.

Syntax

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

Parameters

`unit`	(IN) BCM device number
`flags`	(IN) Destory all objects associated with this endpoint

Description

Destroys all SAT endpoint objects associated with the given endpoint type and free all its allocated resources.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_TIMEOUT`	Unable to obtain resource lock
`BCM_E_INIT`	Module not initialized
`BCM_E_NOT_FOUND`	Attempt to destroy an endpoint which does not exist
`BCM_E_PARAM`	Specified endpoint ID is out of range
`BCM_E_INTERNAL`	Unable to release resource lock / Failed to read or write memory or register

bcm_sat_endpoint_traverse

Traverse the set of SAT endpoints.

Syntax

```
#include <bcm/sat.h>
int bcm_sat_endpoint_traverse(int unit, uint32 flags,
      bcm_sat_endpoint_traverse_cb cb,
      void *user_data);
```

Parameters

`unit`	(IN) BCM device number
`flags`	(IN) Endpoint type(UP/Down MEP) to traverse
`cb`	(IN) Pointer to the callback function
`user_data`	(OUT) Pointer to arbitrary user data to be passed back to the callback

Description

Traverse the set of SAT endpoints associated with the specified endpoint type, calling a specified callback for each one

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_TIMEOUT`	Unable to obtain resource lock
`BCM_E_INIT`	Module not initialized
`BCM_E_PARAM`	No user callback specified / Specified endpoint ID is out of range
`BCM_E_INTERNAL`	Unable to release resource lock
`BCM_E_XXX`	Other codes can be returned by the user callback

MULTI-DEVICE STACK CONTROL

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

bcm_stk_module_max_set
bcm_stk_module_max_get

Maximum group ID set/get

Syntax

```
#include <bcm/stack.h>
int bcm_stk_module_max_set(int unit, uint32 flags, bcm_module_t max_module)
int bcm_stk_module_max_get(int unit, uint32 flags, bcm_module_t *max_module)
```

Parameters

`unit`	(IN) Unit number
`flags`	(IN) Relevant flags
`max_module`	(for "_set", IN) Max module id to be configured
`max_module`	(for "_get", OUT) Max module id configured

Description

Set/Get a maximum module id to improve response time/exclude module id from all reachable calcualtion.

Table 64: Stack Module Max Flags

Flag	*Description*
`BCM_STK_MODULE_MAX_ALL_REACHABLE`	Use this flag in order to configure max module id for all reachable vector calculation

Returns

BCM_E_XXX

bcm_stk_module_control_set
bcm_stk_module_control_get

Set/get control value.

Syntax

```
#include <bcm/stack.h>
int bcm_stk_module_max_set(int unit, uint32 flags, bcm_module_t module,
bcm_stk_module_control_t control, int arg)
int bcm_stk_module_max_set(int unit, uint32 flags, bcm_module_t module,
bcm_stk_module_control_t control, int *arg)
```

Parameters

`unit`	(IN) Unit number
`flags`	(IN) Relevant flags
`module`	(IN) Module ID
`control`	(IN) Relevant control
`arg`	(for "_set", IN) value to be configured
`max_module`	(for "_get", OUT) value configured

Description

Set/get relevant control value per module id

Table 65: bcm_stk_module_control_t

enum	*Description*
bcmStkModuleAllReachableIgnore	Ignore module ID in all reachable vector calculation

Returns

BCM_E_XXX

STATISTICS

A new object has been added in the enumeration of Ingress and Egress Statistic Accounting Objects with a new packets64 field of `bcm_stat_value_t` data structure for 64-bit accumulated packets value.

```
/* Ingress and Egress Statistics Accounting Objects */
typedef enum bcm_stat_object_e {
    ...
    bcmStatObjectIngVxlanDip = 33       /* Ingress Vxlan Dip Object */
} bcm_stat_object_t;

/* Counter Statistics Values */
typedef struct bcm_stat_value_s {
    ...
    uint64 packets64;   /* 64-bit accumulated packets value */
} bcm_stat_value_t;
```

New statistics APIs have been added as following along with the required data types.

bcm_stat_flex_pool_info_multi_get

Retrieves the flex counter details for a given direction

Syntax

```
#include <bcm/stat.h>
int
bcm_stat_flex_pool_info_multi_get(
    int unit,
    bcm_stat_flex_direction_t direction,
    uint32 num_pools,
    uint32 *actual_num_pools,
    bcm_stat_flex_pool_stat_info_t *flex_pool_stat)
```

Parameters

`unit`	(IN) Unit number
`direction`	(IN) Direction
`num_pools`	(IN) Passing a 0, then actual_num_pools will return
`actual_num_pools`	(OUT) Returns actual no of pools
`flex_pool_stat`	(INOUT) array that provides the pool info

Description

This API retrieves the flex counter details for a given direction

Returns

`BCM_E_XXX`

bcm_stat_value_t_init

Initialize a bcm_stat_value_t data structure

Syntax

```
#include <bcm/stat.h>
void
bcm_stat_value_t_init(
    bcm_stat_value_t *stat_value);
```

Parameters

`stat_value`	(INOUT) pointer to the bcm_stat_value_t data to be initialized.

Description

This API initializes the `bcm_stat_value_t` data structure.

Returns

NONE

SWITCH CONTROL

Two new switch control types have been added for `bcm_switch_control_xxx()` functions.

Table 66: Switch Type Values

Value	***Description***	***Arg Value***
`bcmSwitchControlPortConfigInstall`	Install new port configuration. For XCore/Caladan3 device only.	`0x0-0x1fff`
`bcmSwitchMplsPipeTunnelLabelExpSet`	If set, MPLS pipe modes is supported with EXP field set.br In this case exp is taken from the exp field in `bcm_mpls_egress_label_s` struct in the `bcm_mpls_port_add` api.br Otherwise EXP field will be copied from previous label. Default is 0.	`0/1`
`bcmSwitchLoopbackMtuSize`	MTU size in bytes check for packets ingressed on loopback port.	`1-0x3fff`
`bcmSwitchHashVxlanPayloadSelect0`	Set hash control to select VXLAN payload L2/L3 fields for Hash Block A.	• `BCM_HASH_SELECT_INNER_L2` : Inner L2 fields • `BCM_HASH_SELECT_INNER_L3` : Inner L3 fields
`bcmSwitchHashVxlanPayloadSelect1`	Set hash control to select VXLAN payload L2/L3 fields for Hash Block B.	• `BCM_HASH_SELECT_INNER_L2` : Inner L2 fields • `BCM_HASH_SELECT_INNER_L3` : Inner L3 fields
`bcmSwitchCongestionCnmSrcMacNonOui`	Source MAC address in Congestion Notification Message Lower 3 bytes of MAC address to match	
`bcmSwitchCongestionCnmSrcMacOui`	Source MAC address in Congestion Notification Message Upper 3 bytes of MAC address to match	

A new field fwd_reason of bcm_switch_pkt_info_t has been added for bcm_switch_pkt_info_hash_get() function with the newly defined bcm_switch_pkt_hash_info_fwd_reason_t while a new flag of switch packet information has been defined for the flags field of bcm_switch_pkt_info_t.

```
typedef struct bcm_switch_pkt_info_s {
    ...
   bcm_switch_pkt_hash_info_fwd_reason_t fwd_reason; /* Packet forward reason.
*/
} bcm_switch_pkt_info_t;

typedef enum bcm_switch_pkt_hash_info_fwd_reason_e {
    bcmSwitchPktHashInfoFwdReasonUnicast = 0, /* Known unicast packet */
    bcmSwitchPktHashInfoFwdReasonIpmc = 1, /* L3 multicast packet */
    bcmSwitchPktHashInfoFwdReasonL2mc = 2, /* L2 multicast packet */
    bcmSwitchPktHashInfoFwdReasonBcast = 3, /* Broadcast packet */
    bcmSwitchPktHashInfoFwdReasonDlf = 4 /* Destination lookup failed packet */
} bcm_switch_pkt_hash_info_fwd_reason_t;
```

Table 67: BCM Switch packet information flags

Name	Purpose
BCM_SWITCH_PKT_INFO_HASH_UDP_SOURCE_PORT	Retrieve Entropy Value of VxLan packet.

Several statistics objects as below have been added for bcm_switch_object_count_get() and bcm_switch_object_count_multi_get() functions.

```
typedef enum bcm_switch_object_e {
    ...
    bcmSwitchObjectL3RouteV4RoutesMax,       /* Maximum number of v4 routes
possible */
   bcmSwitchObjectL3RouteV4RoutesFree,     /* Maximum number of v4 routes that
can
                                           be added in current state */
    bcmSwitchObjectL3RouteV4RoutesUsed,      /* Used count of v4 routes */
    bcmSwitchObjectL3RouteV6Routes64bMax,    /* Maximum number of 64bv6 routes
                                           possible */
    bcmSwitchObjectL3RouteV6Routes64bFree,   /* Maximum number of 64bV6 routes
that
                                           can be added in current state */
    bcmSwitchObjectL3RouteV6Routes64bUsed,   /* Used count of 64bv6 routes */
    bcmSwitchObjectL3RouteV6Routes128bMax,   /* Maximum number of 128bV6 routes
                                           possible */
   bcmSwitchObjectL3RouteV6Routes128bFree, /* Maximum number of 128bV6 routes
that
                                           can be added in current state */
    bcmSwitchObjectL3RouteV6Routes128bUsed, /* Used count of 128bv6 routes */
   bcmSwitchObjectL3RouteTotalUsedRoutes,  /* Sum of v4 + 64bv6 + 128bv6 routes
*/
    bcmSwitchObjectIpmcHeadTableFree,        /* Number of free entries in the
replication head
```

```
                                           table */
} bcm_switch_object_t;
```

New switch control APIs added in this release are listed as following:

bcm_switch_l3_protocol_group_set
bcm_switch_l3_protocol_group_get

Assign/get protocol groups for multiple mymac termination

Syntax

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

typedef uint32 bcm_l3_protocol_group_id_t;

extern int bcm_switch_l3_protocol_group_set(
    int unit,
    uint32 group_members,
    bcm_l3_protocol_group_id_t group_id);

extern int bcm_switch_l3_protocol_group_get(
    int unit,
    uint32 *group_members,
    bcm_l3_protocol_group_id_t *group_id);
```

Parameters

`unit`	(IN) Unit number.
`group_members`	(IN) (for set) The protocols to be assigned to the group.
`group_members`	(INOUT) (for get) The protocols assigned to the group.
`group_id`	(IN) (for set) The group to which the l3 protocols are to be assigned.
`group_id`	(INOUT) (for get) The group to which the l3 protocols are assigned.

Description

When using multiple mymac termination feature, L3 protocols can be divided into different groups. The group is part of the key used to decide whether the packet will be l2 terminated. Use this api to assign the different l3 protocols into groups. The default group for all protocols is 0, and protocls in re-set groups that weren't reassigned would be automatically reassigned to group 0.

For the get api, the input can be either the group members, and the group number will be returned, or the input can be the group number, and all members assigned to it will be returned. Either way, the other one must be -1 (`MAX_UINT32` for flags and `MAX_UINT8` for `group_id`).

The protocol are given in the form of flags. The flags are:

Table 68: BCM_SWITCH_L3_PROTOCOL_GROUP_* flags

Name	Description	Value
BCM_SWITCH_L3_PROTOCOL_GROUP_NONE	No protocols in this group.	0x0000
BCM_SWITCH_L3_PROTOCOL_GROUP_IPV4	Add IPv4 to the group.	0x0001
BCM_SWITCH_L3_PROTOCOL_GROUP_IPV6	Add IPv6 to the group.	0x0002
BCM_SWITCH_L3_PROTOCOL_GROUP_ARP	Add ARP to the group.	0x0004
BCM_SWITCH_L3_PROTOCOL_GROUP_MPLS	Add MPLS to the group.	0x0008
BCM_SWITCH_L3_PROTOCOL_GROUP_MIM	Add MiM to the group.	0x0010
BCM_SWITCH_L3_PROTOCOL_GROUP_TRILL	Add trill to the group.	0x0020
BCM_SWITCH_L3_PROTOCOL_GROUP_FCOE	Add FCOE to the group.	0x0040

Returns

BCM_E_xxx

bcm_switch_profile_mapping_t_init

Initialize a bcm_switch_profile_mapping_t structure

Syntax

```
#include <bcm/switch.h>
void
bcm_switch_profile_mapping_t_init(bcm_switch_profile_mapping_t
*profile_mapping);
```

Parameters

profile_mapping	(IN/OUT)	profile mapping structure.

Description

Initialize a profile mapping structure to default values. This structure is used for creating profile to gport mapping.

Returns

void

bcm_switch_hash_entry_create_qset

Create a blank flex hash entry.

Syntax

```
#include <bcm/switch.h>
int
bcm_switch_hash_entry_create_qset(
    int unit,
    bcm_field_qset_t qset,
    bcm_hash_entry_t *entry);
```

Parameters

`unit`	(IN) Unit number.
`qset`	(IN) <UNDEF>
`entry`	(OUT) <UNDEF>

Description

This API, unlike , takes a `bcm_field_qset_t` INPUT instead of a `bcm_field_group_t` INPUT. The qset should be filled with the UDF APIs defined in (UDF Resources Management is new to this release) (page 170)

Returns

`BCM_E_xxx`

TRILL MANAGEMENT

New trill port flags for `bcm_trill_port_t` have been defined along with the applicable data structure `bcm_trill_rbridge_t` and corresponding APIs.

Table 69: TRILL port flags

Name	*Purpose*
BCM_TRILL_PORT_TUNNEL	Create unicast Rbridge without transit
BCM_TRILL_PORT_INGRESS	Create Ingress Rbridge
BCM_TRILL_PORT_EGRESS	Create Egress Rbridge

```
typedef struct bcm_trill_rbridge_s {
    uint32 flags;            /* BCM_TRILL_RBRIDGE_xxx. */
    bcm_trill_name_t name;   /* RBridge Nickname. */
    bcm_if_t egress_if;      /* TRILL egress object. */
    int mtu;                 /* TRILL port MTU. */
    int hopcount;            /* Hopcount for TRILL. */
} bcm_trill_rbridge_t;
```

New APIs of trill management have been added in this release with the data types.

- The `bcm_trill_multicast_adjacency_t` structure describes a TRILL multicast adjacency entry used for performing trill adjacency check

```
typedef struct bcm_trill_multicast_adjacency_s {
    bcm_gport_t port;        /* Match Trill network port. */
    uint32 flags;           /* BCM_TRILL_MULTICAST_ADJACENCY_xxx. */
    bcm_if_t encap_intf;    /* Neighbor interface. */
} bcm_trill_multicast_adjacency_t;
```

bcm_trill_multicast_adjacency_add

Add Multicast adjacency entry

Syntax

```
#include <bcm/trill.h>
int bcm_trill_multicast_adjacency_add(int unit,
       bcm_trill_multicast_adjacency_t *trill_multicast_adjacency);
```

Parameters

`unit`	(IN) BCM unit
`trill_multicast_adjacency`	(INOUT) Trill multicast adjacency entry

Description

Add Trill Multicast adjacency entry

bcm_trill_multicast_adjacency_delete

Delete Multicast adjacency entry

Syntax

```
#include <bcm/trill.h>
int bcm_trill_multicast_adjacency_delete(int unit,
 bcm_trill_multicast_adjacency_t *trill_multicast_adjacency);
```

Parameters

`unit`	(IN) BCM unit
`trill_multicast_adjacency`	(IN) Trill multicast adjacency entry

Description

Delete Trill Multicast adjacency entry

bcm_trill_multicast_adjacency_delete_all

Delete all Multicast adjacency entry

Syntax

```
#include <bcm/trill.h>
int bcm_trill_multicast_adjacency_delete_all(int unit);
```

Parameters

`unit`	(IN) BCM unit

Description

Delete all Trill Multicast adjacency entry

bcm_trill_multicast_adjacency_traverse

Traverse all valid TRILL Multicast adjacency entries and call supplied callback routine.

Syntax

```
#include <bcm/trill.h>
int bcm_trill_multicast_adjacency_traverse(int unit,
    bcm_trill_multicast_adjacency_traverse_cb cb,
    void *user_data);
```

Parameters

`unit`	(IN) BCM unit
`flags`	(IN) User callback function, called once per TRILL entry
`user_data`	(IN) Cookie

Description

Traverse all valid TRILL Multicast adjacency entries and call supplied callback routine.

Returns

```
BCM_E_XXX
```

bcm_trill_rbridge_entry_add

bcm_trill_rbridge_entry_add Add TRILL Rbridge entry.

Syntax

```
#include <bcm/trill.h>
int bcm_trill_rbridge_entry_add(int unit,
bcm_trill_rbridge_t *trill_entry);
```

Parameters

`unit`	(IN) BCM unit
`trill_entry`	(IN/OUT) TRILL Rbridge entry

Description

Add TRILL Rbridge entry.

Returns

```
BCM_E_XXX
```

bcm_trill_rbridge_entry_delete

bcm_trill_rbridge_entry_delete Add TRILL Rbridge entry.

Syntax

```
#include <bcm/trill.h>
int bcm_trill_rbridge_entry_delete(int unit,
bcm_trill_rbridge_t *trill_entry);
```

Parameters

`unit`	(IN) BCM unit
`trill_entry`	(IN) TRILL Rbridge entry

Description

Delete TRILL Rbridge entry.

Returns

```
BCM_E_XXX
```

bcm_trill_rbridge_entry_get

bcm_trill_rbridge_entry_get Add TRILL Rbridge entry.

Syntax

```
#include <bcm/trill.h>
int bcm_trill_rbridge_entry_get(int unit,
bcm_trill_rbridge_t *trill_entry);
```

Parameters

unit	(IN) BCM unit
trill_entry	(IN/OUT) TRILL Rbridge entry

Description

Get TRILL Rbridge entry.

Returns

BCM_E_XXX

TRUNKING (LINK AGGREGATION)

A new trunk flag has been added in this release as below.

Table 70: BCM Trunk Flags

Name	***Purpose***
BCM_TRUNK_FLAG_MEMBER_SORT	If set, set or add trunk members to hardware in the sorted way.

VLAN MANAGEMENT

A new field vp_mc_ctrl of bcm_vlan_control_vlan_t data structure has been added as well as its data type of bcm_vlan_vp_mc_ctrl_t and BCM_VLAN_CONTROL_VLAN_VP_MCAST_MASK of VLAN Control Valid Field Mask to control VP replication of the VLAN.

Table 71: VLAN Control Valid Field Mask

VLAN Control Valid Field Mask	***Meaning***
BCM_VLAN_CONTROL_VLAN_VP_MCAST_MASK	Enable .vp_mc_ctrl field.

```
typedef struct {
    ...
    bcm_vlan_vp_mc_ctrl_t   vp_mc_ctrl;   /* VP replication control, Auto,
Enable, Disable */
} bcm_vlan_control_vlan_t;

typedef enum bcm_vlan_vp_mc_ctrl_e {
    bcmVlanVPMcControlAuto = 0,     /* VP Multicast replication auto control */
    bcmVlanVPMcControlEnable = 1,   /* Enable VP Multicast replication */
    bcmVlanVPMcControlDisable = 2   /* Disable VP Multicast replication */
} bcm_vlan_vp_mc_ctrl_t;
```

Two new fields egress_failover_id and egress_failover_port_id of bcm_vlan_port_t data structure have been added.

```
typedef struct bcm_vlan_port_s {
    ...
    bcm_failover_t egress_failover_id;   /* Failover object index for Egress 
Protection */
    bcm_gport_t egress_failover_port_id; /* Failover VLAN Port Identifier for 
Egress Protection */
} bcm_vlan_port_t;
```

New VLAN stat sync get APIs and stat counter sync get APIs are listed below.

bcm_vlan_stat_counter_sync_get

Force an immediate counter update and retrieve the specified statistic to the indicated value for the specified VLAN.

Syntax

```
#include <bcm/vlan.h>
int
bcm_vlan_stat_counter_sync_get(
    int                 unit,
    bcm_vlan_t          vlan,
    bcm_vlan_stat_t     stat,
    uint32              num_entries,
    uint32              *counter_indexes,
    bcm_stat_value_t    *counter_values);
```

Parameters

unit	(IN) Unit number.
vlan	(IN) VLAN ID
stat	(IN) Type of the counter to retrieve
num_entries	(IN) Number of counter Entries
counter_indexes	(IN) Pointer to Counter indexes entries
counter_values	(OUT) Pointer to counter values

Description

Similar to bcm_vlan_stat_counter_get(), value returned is software accumulated counter synced with the hardware counter.

Returns

```
BCM_E_xxx
```

bcm_vlan_stat_sync_get
bcm_vlan_stat_sync_get32

Force an immediate counter update and retrieve per-VLAN statistics from the chip.

Syntax

```
#include <bcm/vlan.h>
int
bcm_vlan_stat_sync_get(
    int unit,
    bcm_vlan_t vlan,
    bcm_cos_t cos,
    bcm_vlan_stat_t stat,
    uint64 *val);

int
bcm_vlan_stat_sync_get32(
    int unit,
    bcm_vlan_t vlan,
    bcm_cos_t cos,
    bcm_vlan_stat_t stat,
    uint32 *val);
```

Parameters

unit	(IN) Unit number.
vlan	(IN) VLAN ID.
cos	(IN) CoS or priority
stat	(IN) Type of the counter to retrieve.
val	(OUT) Pointer to a counter value.

Description

Similar to bcm_vlan_stat_get(), value returned is software accumulated counter synced with the hardware counter

Returns

BCM_E_XXX

bcm_vlan_translate_stat_counter_sync_get

Force an immediate counter update and retrieve counter values from ingress VLAN translation table.

Syntax

```
#include <bcm/vlan.h>
int
bcm_vlan_translate_stat_counter_sync_get(
    int                       unit,
    bcm_gport_t               port,
    bcm_vlan_translate_key_t key_type,
    bcm_vlan_t                outer_vlan,
    bcm_vlan_t                inner_vlan,
    bcm_vlan_stat_t           stat,
    uint32                    num_entries,
    uint32                    *counter_indexes,
    bcm_stat_value_t          *counter_values)
```

Parameters

`unit`	(IN) Unit number.
`port`	(IN) GPORT (global port) identifier
`key_type`	(IN) VLAN translation key type
`outer_vlan`	(IN) Outer VLAN ID or tag
`inner_vlan`	(IN) Inner VLAN ID or tag
`stat`	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_vlan_translate_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_xxx`

bcm_vlan_translate_stat_sync_get bcm_vlan_translate_stat_sync_get32

Force an immediate counter update and retrieve ingress VLAN translation statistics from the chip.

Syntax

```
#include <bcm/vlan.h>
int bcm_vlan_translate_stat_sync_get(int unit, bcm_gport_t port,
     bcm_vlan_translate_key_t key_type,
     bcm_vlan_t outer_vlan,
     bcm_vlan_t inner_vlan,
     bcm_vlan_stat_t stat,
     uint64 *val);

int bcm_vlan_translate_stat_sync_get32(int unit, bcm_gport_t port,
       bcm_vlan_translate_key_t key_type,
       bcm_vlan_t outer_vlan,
       bcm_vlan_t inner_vlan,
       bcm_vlan_stat_t stat,
       uint32 *val);
```

Parameters

`unit`	(IN) Unit number.
`port`	(IN) Generic port
`key_type`	(IN) Key Type : bcmVlanTranslateKey*
`outer_vlan`	(IN) Packet outer VLAN ID
`inner_vlan`	(IN) Packet inner VLAN ID
`stat`	(IN) Type of the counter to retrieve.
`val`	(OUT) Pointer to a counter value.

Description

Similar to `bcm_vlan_translate_stat_get(),` value returned is software accumulated counter synced with the hardware counter

Returns

`BCM_E_xxx`

bcm_vlan_translate_egress_stat_counter_sync_get

Force an immediate counter update and retrieve counter values from egress VLAN translation table.

Syntax

```
#include <bcm/vlan.h>
int
bcm_vlan_translate_egress_stat_counter_sync_get(
    int                 unit,
    int                 port_class,
    bcm_vlan_t          outer_vlan,
    bcm_vlan_t          inner_vlan,
    bcm_vlan_stat_t     stat,
    uint32              num_entries,
    uint32              *counter_indexes,
    bcm_stat_value_t    *counter_values)
```

Parameters

`unit`	(IN) Unit number.
`port_class`	(IN) port_class
`outer_vlan`	(IN) Outer VLAN ID or tag
`inner_vlan`	(IN) Inner VLAN ID or tag
`stat`	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_vlan_translate_egress_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

`BCM_E_xxx`

bcm_vlan_translate_egress_stat_sync_get bcm_vlan_translate_egress_stat_sync_get32

Force an immediate counter update and retrieve egress VLAN translation statistics from the chip.

Syntax

```
#include <bcm/vlan.h>
int bcm_vlan_translate_egress_stat_sync_get(int unit, int port_class,
    bcm_vlan_t outer_vlan,
    bcm_vlan_t inner_vlan,
    bcm_vlan_stat_t stat,
    uint64 *val);

int bcm_vlan_translate_egress_stat_sync_get32(int unit, int port_class,
      bcm_vlan_t outer_vlan,
      bcm_vlan_t inner_vlan,
      bcm_vlan_stat_t stat,
      uint32 *val);
```

Parameters

unit	(IN) Unit number.
port_class	(IN) Group ID of ingress port
outer_vlan	(IN) Packet outer VLAN ID
inner_vlan	(IN) Packet inner VLAN ID
stat	(IN) Type of the counter to retrieve.
val	(OUT) Pointer to a counter value.

Description

Similar to bcm_vlan_translate_egress_stat_get(), value returned is software accumulated counter synced with the hardware counter.

Returns

BCM_E_xxx

VXLAN MANAGEMENT

A new flag has been added for bcm_vxlan_tunnel_terminator_update() API while BCM_VXLAN_PORT_MATCH_VN_ID has been deprecated from VXLAN Port Match Criteria in this release.

Table 72: VXLAN Port Match Criteria

Name	***Purpose***
BCM_VXLAN_PORT_MATCH_VN_ID	VXLAN VN_ID

- bcm_vxlan_tunnel_terminator_update API provides a trigger mechanism to update multicast BUD state per VXLAN Tunnel terminator.

Table 73: VXLAN Tunnel terminator Multicast flags

name	***heading Purpose***
BCM_TUNNEL_TERM_TUNNEL_DEACTIVATE	Deactivate Tunnel Terminator

A new flag BCM_VXLAN_PORT_ENABLE_VLAN_CHECKS that Enable VLAN checks for VXLAN port has been added for data type bcm_vxlan_port_t.

- The bcm_vxlan_port_t structure describes a VXLAN virtual port used for creating VXLAN Network endpoints.

Table 74: VXLAN port flags

Name	***Purpose***
BCM_VXLAN_PORT_ENABLE_VLAN_CHECKS	Enable VLAN Checks for VXLAN Port

A new field `protocol_pkt` of `bcm_vxlan_vpn_config_t` data structure has been added for the configuring of VXLAN vpn in this release.

```
typedef struct bcm_vxlan_vpn_config_s {
    ...
    bcm_vlan_protocol_packet_ctrl_t protocol_pkt;
} bcm_vxlan_vpn_config_t;
```

Table 75: VXLAN Port Match Criteria

Name	Purpose
BCM_VXLAN_PORT_DEFAULT	VXLAN Default Network Port

The new APIs with regard to VXLAN Management are listed as following as well as the data type `bcm_vxlan_dip_stat_t`:

```
/* Ingress statistics maintained per Vxlan Dip. */
typedef enum bcm_vxlan_dip_stat_e {
    bcmVxlanDipInPackets = 0,    /* Packets that ingress on the Vxlan Dip . */
    bcmVxlanDipInBytes = 1       /* Bytes that ingress on the Vxlan Dip . */
} bcm_vxlan_dip_stat_t;
```

bcm_vxlan_stat_enable_set

Enable/Disable collection of statistics on the indicated port and vpn

Syntax

```
#include <bcm/vxlan.h>
int
bcm_esw_vxlan_stat_enable_set(
    int              unit,
    bcm_gport_t      port,
    bcm_vpn_t        vpn,
    int              enable);
```

Parameters

unit	(IN) Unit number.
port	(IN) vxlan vp index
vpn	(IN) vxlan vpn index
enable	(IN) Non-zero to enable counter collection, zero to disable.

Description

This API will initialize statistic collection for the given vp/vpn (enable=TRUE) or release the HW resources used for the tracking the statistics (enable=FALSE). The below table mentions the flex stat actions for the input vxlan parameters.

Table 76:

	VxlanPort	Vpn	flex stat Action
1.	ValidPort	-1	Port stat Enable/Disable
2.	-1	Validvpn	Vpn stat Enable/Disable
3.	ValidPort	ValidVpn	Port+Vpn Enable/Disable
4.	InvalidPort	InvalidVpn	return `BCM_E_PARAM`

Returns

`BCM_E_xxx`

bcm_vxlan_dip_stat_attach

Attach counters entries to a given VXLAN DIP.

Syntax

```
#include <bcm/vxlan.h>
int
bcm_vxlan_dip_stat_attach (
    int             unit,
    bcm_ip_t        vxlan_dip;
    uint32          stat_counter_id);
```

Parameters

`unit`	(IN) Unit number.
`vxlan_dip`	(IN) Vxlan DIP
`stat_counter_id`	(IN) Stat Counter ID

Description

This API will attach counters entries to a given VXLAN DIP.

Returns

`BCM_E_xxx`

bcm_vxlan_dip_stat_detach

Detach counters entries to a given VXLAN DIP.

Syntax

```
#include <bcm/vxlan.h>
int
bcm_vxlan_dip_stat_detach (
    int            unit,
    bcm_ip_t       vxlan_dip;);
```

Parameters

`unit`	(IN) Unit number.
`vxlan_dip`	(IN) Vxlan DIP

Description

This API will detach counters entries to a given VXLAN DIP.

Returns

```
BCM_E_xxx
```

bcm_vxlan_dip_stat_counter_get

Get the specified counter statistic for a given VXLAN DIP.

Syntax

```
#include <bcm/vxlan.h>
int
bcm_vxlan_dip_stat_counter_get(
    int                     unit,
    bcm_ip_t                vxlan_dip;
    bcm_vxlan_dip_stat_t    stat,
    uint32                  num_entries,
    uint32                  *counter_indexes,
    bcm_stat_value_t        *counter_values);
```

Parameters

`unit`	(IN) Unit number.
`vxlan_dip`	(IN) Vxlan DIP.
`stat`	(IN) Type of the counter to retrieve.
`num_entries`	(IN) Number of counter Entries.
`counter_indexes`	(IN) Pointer to Counter indexes entries.
`counter_values`	(OUT) Pointer to counter values.

Description

This API will get the specified counter statistic for a given VXLAN DIP.

Returns

```
BCM_E_xxx
```

bcm_vxlan_dip_stat_counter_set

Set the specified counter statistic for a given VXLAN DIP.

Syntax

```
#include <bcm/vxlan.h>
int
bcm_vxlan_dip_stat_counter_set(
    int                       unit,
    bcm_ip_t                  vxlan_dip;
    bcm_vxlan_dip_stat_t      stat,
    uint32                    num_entries,
    uint32                    *counter_indexes,
    bcm_stat_value_t          *counter_values);
```

Parameters

`unit`	(IN) Unit number.
`vxlan_dip`	(IN) Vxlan DIP.
`stat`	(IN) Type of the counter to retrieve.
`num_entries`	(IN) Number of counter Entries.
`counter_indexes`	(IN) Pointer to Counter indexes entries.
`counter_values`	(IN) Pointer to counter values.

Description

This API will set the specified counter statistic for a given VXLAN DIP.

Returns

```
BCM_E_xxx
```

bcm_vxlan_dip_stat_multi_get

Get 64-bit counter value for multiple VXLAN DIP statistic types.

Syntax

```
#include <bcm/vxlan.h>
int
bcm_vxlan_dip_stat_multi_get(
    int                       unit,
    bcm_ip_t                  vxlan_dip,
    int                       nstat,
    bcm_vxlan_dip_stat_t      *stat_arr,
    uint64                    *value_arr);
```

Parameters

`unit`	(IN) Unit number.
`vxlan_dip`	(IN) Vxlan DIP.
`nstat`	(IN) Number of elements in stat array.
`stat_arr`	(IN) Collected statistics descriptors array.
`value_arr`	(OUT) Collected counters values.

Description

This API will get 64-bit counter value for multiple VXLAN DIP statistic types.

Returns

BCM_E_xxx

bcm_vxlan_dip_stat_multi_get32

Get 32-bit counter value for multiple VXLAN DIP statistic types.

Syntax

```
#include <bcm/vxlan.h>
int
bcm_vxlan_dip_stat_multi_get32(
    int                       unit,
    bcm_ip_t                  vxlan_dip,
    int                       nstat,
    bcm_vxlan_dip_stat_t      *stat_arr,
    uint32                    *value_arr);
```

Parameters

`unit`	(IN) Unit number.
`vxlan_dip`	(IN) Vxlan DIP.
`nstat`	(IN) Number of elements in stat array.
`stat_arr`	(IN) Collected statistics descriptors array.
`value_arr`	(OUT) Collected counters values.

Description

This API will get 32-bit counter value for multiple VXLAN DIP statistic types.

Returns

BCM_E_xxx

bcm_vxlan_dip_stat_multi_set

set 64-bit counter value for multiple VXLAN DIP statistic types.

Syntax

```
#include <bcm/vxlan.h>
int
bcm_vxlan_dip_stat_multi_set(
    int                     unit,
    bcm_ip_t                vxlan_dip,
    int                     nstat,
    bcm_vxlan_dip_stat_t    *stat_arr,
    uint64                  *value_arr);
```

Parameters

`unit`	(IN) Unit number.
`vxlan_dip`	(IN) Vxlan DIP.
`nstat`	(IN) Number of elements in stat array.
`stat_arr`	(IN) Collected statistics descriptors array.
`value_arr`	(IN) Collected counters values.

Description

This API will set 64-bit counter value for multiple VXLAN DIP statistic types.

Returns

`BCM_E_xxx`

bcm_vxlan_dip_stat_multi_set32

set 32-bit counter value for multiple VXLAN DIP statistic types.

Syntax

```
#include <bcm/vxlan.h>
int
bcm_vxlan_dip_stat_multi_set32(
    int                     unit,
    bcm_ip_t                vxlan_dip,
    int                     nstat,
    bcm_vxlan_dip_stat_t    *stat_arr,
    uint64                  *value_arr);
```

Parameters

`unit`	(IN) Unit number.
`vxlan_dip`	(IN) Vxlan DIP.
`nstat`	(IN) Number of elements in stat array.
`stat_arr`	(IN) Collected statistics descriptors array.
`value_arr`	(IN) Collected counters values.

Description

This API will set 32-bit counter value for multiple VXLAN DIP statistic types.

Returns

```
BCM_E_xxx
```

bcm_vxlan_dip_stat_id_get

Retrieve associated stat counter for a given VXLAN DIP.

Syntax

```
#include <bcm/vxlan.h>
int
bcm_vxlan_dip_stat_id_get(
    int                     unit,
    bcm_ip_t                vxlan_dip,
    bcm_vxlan_dip_stat_t    stat,
    uint32                  *stat_counter_id);
```

Parameters

`unit`	(IN) Unit number.
`vxlan_dip`	(IN) Vxlan DIP.
`stat`	(IN) Type of the counter to retrieve.
`stat_counter_id`	(IN) Stat Counter ID.

Description

This API will retrieve associated stat counter for a given VXLAN DIP.

Returns

```
BCM_E_xxx
```

bcm_vxlan_stat_counter_sync_get

Force an immediate counter update and retrieve counter statistic values for a vxlan vp/vpn index.

Syntax

```
#include <bcm/vxlan.h>
int
bcm_vxlan_stat_counter_sync_get(
    int                unit,
    bcm_gport_t       port,
    bcm_vpn_t         vpn,
    bcm_vxlan_stat_t     stat,
    uint32             num_entries,
    uint32             *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

unit	(IN) Unit number.
port	(IN) vxlan vp index
vpn	(IN) vxlan vpn index
stat	(IN) Vxlan counter stat types.
num_entries	(IN) Number of counter Entries
counter_indexes	(IN) Pointer to Counter indexes entries
counter_values	(OUT) Pointer to counter values

Description

Similar to `bcm_vxlan_stat_counter_get(),` value returned is software accumulated counter synced with the hardware counter.

Returns

BCM_E_xxx

UDF RESOURCES MANAGEMENT IS NEW TO THIS RELEASE

The new BCM UDF APIs allow application developers to manage the UDF resources in StrataXGS devices.

UDF RESOURCES

Historically, in StrataXGS devices the UDF resources were exclusively used by the Field Processor for flexible classification of user defined patterns in the incoming packets. The APIs for managing these resources were hence defined within FIELD module.

The UDF selection logic is now being used by many other applications apart from the ContentAware Engine. For example, BCM56850 uses the UDF selected chunks for flexible hashing using layer 4 payload data.

For the purpose of centralized management of these shared resources, these new set of APIs are defined. These APIs will supersede the existing field data qualifier management APIs. These APIs will control the management of the UDF resources and the applications (like field, hashing etc..) will have APIs that use these objects for programming the switch for the desired functionality.

API `bcm_udf_create` takes "layer" (base offset), "start" (relative offset) and "width" (window of udf) inputs through `bcm_udf_t` input and allocates an opaque `bcm_udf_id_t` handle to the UDF object thus created. This API can optionally take "hints" from the user through bcm_udf_alloc_hints_t to reserve the hardware offset chunks in a manner that is best suitable for other applications in the pipeline - Vlan Stage Field Processor, Ingress Stage Field Processor, Flexible hashing entries etc...

A udf object will only be functional when it is associated with a packet format entry. A single udf object can be associated with multiple packet format entries and similarly a sinlge packet format entry can be associated with multiple udf objects. And when the udf is dissociated from all the packet format entries, the entry will be removed from hardware but the object would be removed from software only when it is destroyed using bcm_udf_destroy API.

API bcm_udf_pkt_format_create creates a packet format matching entry to qualify the packets for UDF selection logic. This API returns a bcm_udf_pkt_format_entry_t handle to refer to the packet format entry thus created.

In case of ambiguous packet format matching entries, the entry with the higher priority will take precedence and the udf selection will happen according to the associated udf objects.

Since the number of udf objects are limited by the hardware, users can create shared objects that would use the same hardware offsets as the other udf but they need to be associated with different packet format entries.

These APIs supercede the bcm_field_data_qualifier_* APIs defined here: The intermixed usage is not supported.

UDF DATA STRUCTURES

- The UDF.

```
typedef struct bcm_udf_s {
    uint32 flags;           /* See BCM_UDF_F_XXX */
    bcm_udf_layer_t layer;  /* UDF base offset. bcmUdfLayerXxx */
    int start;              /* Relative offset in bits (from the 'layer') in
the
                   packet */
    uint32 width;           /* width of data in bits to extract */
} bcm_udf_t;
```

- UDF allocation hints.

```
typedef struct bcm_udf_alloc_hints_s {
    uint32 flags;               /* See BCM_UDF_CREATE_O_xxx */
    bcm_udf_id_t shared_udf;    /* Used when BCM_UDF_CREATE_O_SHARED_HWID is
set */
    bcm_field_qset_t qset;      /* Field group qset for optimal H/W resource
                   allocation */
} bcm_udf_alloc_hints_t;
```

- Packet format based UDF specification structure.

```
typedef struct bcm_udf_pkt_format_info_s {
    int prio;                       /* Priority of the UDF */
    bcm_ethertype_t ethertype;      /* Ethertype in the packet */
    bcm_ethertype_t ethertype_mask; /* Ethertype Mask */
    uint8 ip_protocol;              /* IP protocol field in the packet */
    uint8 ip_protocol_mask;          /* IP protocol Mask */
     uint16 l2;                           /* L2 packet format.
(BCM_PKT_FORMAT_L2_XXX) */
    uint16 vlan_tag;                 /* Vlan tag format.
                       (BCM_PKT_FORMAT_VLAN_XXX) */
    uint16 outer_ip;                 /* Outer IP header type.
                       (BCM_PKT_FORMAT_IP_XXX) */
    uint16 inner_ip;                 /* Inner IP header type.
                       (BCM_PKT_FORMAT_IP_XXX) */
     uint16 tunnel;                              /* Tunnel type.
(BCM_PKT_FORMAT_TUNNEL_XXX) */
    uint16 mpls;                    /* Mpls labels. (BCM_PKT_FORMAT_MPLS_XXX)
*/
    uint16 fibre_chan_outer;         /* Fibre Channel outer header type.
                       (BCM_PKT_FORMAT_FIBRE_CHAN_XXX) */
    uint16 fibre_chan_inner;         /* Fibre Channel inner header
                       type.(BCM_PKT_FORMAT_FIBRE_CHAN_XXX) */
    uint16 higig;                    /* HIGIG packet
                       format.(BCM_PKT_FORMAT_HIGIG_XXX) */
    uint16 vntag;                    /* NIV packet
                       format.(BCM_PKT_FORMAT_VNTAG_XXX) */
    uint16 etag;                     /* Extender tag packet
                       format.(BCM_PKT_FORMAT_ETAG_XXX) */
    uint16 cntag;                    /* CNTAG packet
                       format.(BCM_PKT_FORMAT_CNTAG_XXX) */
    uint16 icnm;                     /* ICNM packet
                       format.(BCM_PKT_FORMAT_ICNM_XXX) */
} bcm_udf_pkt_format_info_t;
```

Table 77: Flags to control UDF extraction

flags	Description
BCM_UDF_F_ADJUST_IP4_OPTIONS	Adjust the UDF selection to ignore IPv4 options.
BCM_UDF_F_ADJUST_IP6_OPTIONS	Adjust the UDF selection to ignore IPv6 options.

Table 78: Options to bcm_udf_create

flags	Description
BCM_UDF_CREATE_O_NONE	Default.
BCM_UDF_CREATE_O_WITHID	UDF Id from user will be used.
BCM_UDF_CREATE_O_REPLACE	Replace UDF.
BCM_UDF_CREATE_O_FLEXHASH	Hint to the API so the udf allocation is adjusted to be used by the flexible hashing entries.
BCM_UDF_CREATE_O_FIELD_INGRESS	Hint to the API so the udf allocation is adjusted to be used by Ingress Stage FP groups.
BCM_UDF_CREATE_O_FIELD_LOOKUP	Hint to the API so the udf allocation is adjusted to be used by Lookup Stage FP groups.
BCM_UDF_CREATE_O_SHARED_HWID	Use same hardware chunk position as the "shared_udf".

Table 79: Options to bcm_udf_pkt_format_create

flags	*Description*
BCM_UDF_PKT_FORMAT_CREATE_O_NONE	Default.
BCM_UDF_PKT_FORMAT_CREATE_O_WITHID	Packet format Id from the user will be used.
BCM_UDF_PKT_FORMAT_CREATE_O_REPLACE	Replace packet format entry.

APIS TO MANAGE THE UDF RESOURCES

During bcm_init, The udf module will be initialized (or reinitialized, if warm-booting) before the field module is initialized. This is required, at least in case of warm reloading because the udf objects which are recovered in UDF module are needed for appropriate object recovery while recovering field groups/entries.

bcm_udf_t_init

Initializes the UDF structure

Syntax

```
#include <bcm/udf.h>
void
bcm_udf_t_init(
    bcm_udf_t *udf_info);
```

Parameters

udf_info	(IN/OUT) Pointer to

Description

Initializes the UDF structure

Returns

BCM_E_xxx

bcm_udf_alloc_hints_t_init

Initializes the UDF alloc hints structure

Syntax

```
#include <bcm/udf.h>
void
bcm_udf_alloc_hints_t_init(
    bcm_udf_alloc_hints_t *udf_hints);
```

Parameters

`udf_hints`	(IN/OUT) Pointer to

Description

Initializes the UDF alloc hints structure

Returns

```
BCM_E_xxx
```

bcm_udf_pkt_format_info_t_init

Initialize the UDF packet format structure

Syntax

```
#include <bcm/udf.h>
void
bcm_udf_pkt_format_info_t_init(
    bcm_udf_pkt_format_info_t *pkt_format);
```

Parameters

`pkt_format`	(IN/OUT) Pointer to

Description

Initialize the UDF packet format structure

Returns

```
BCM_E_xxx
```

bcm_udf_init

Initialize UDF module

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_init(
    int unit);
```

Parameters

`unit`	(IN) Unit number.

Description

Initialize UDF module

Returns

BCM_E_NONE	UDF module initialized successfully.
BCM_E_UNIT	Invalid BCM Unit number.
BCM_E_UNAVAIL	Feature not supported.
BCM_E_XXX	Standard error code.

bcm_udf_detach

Detach UDF module

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_detach(
    int unit);
```

Parameters

unit	(IN) Unit number.

Description

Detach UDF module

Returns

BCM_E_NONE	UDF module detached successfully.
BCM_E_UNIT	Invalid BCM Unit number.
BCM_E_INIT	UDF module not initialized.
BCM_E_UNAVAIL	Feature not supported.
BCM_E_XXX	Standard error code.

bcm_udf_create

Creates a UDF object

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_create(
    int unit,
    bcm_udf_alloc_hints_t *hints,
    bcm_udf_t *udf_info,
    bcm_udf_id_t *udf_id);
```

Parameters

unit	(IN) Unit number.
hints	(IN) Hints to UDF allocator
udf_info	(IN) UDF structure
udf_id	(IN/OUT) UDF ID

Description

Returns

BCM_E_NONE	UDF created successfully.
BCM_E_EXISTS	Entry already exists.
BCM_E_FULL	UDF table full.
BCM_E_UNIT	Invalid BCM Unit number.
BCM_E_INIT	UDF module not initialized.
BCM_E_UNAVAIL	Feature not supported.
BCM_E_XXX	Standard error code.

bcm_udf_get

Fetches the UDF object created in the system

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_get(
    int unit,
    bcm_udf_id_t udf_id,
    bcm_udf_t *udf_info);
```

Parameters

unit	(IN) Unit number.
udf_id	(IN) UDF Object ID
udf_info	(OUT) UDF info structure

Description

Fetches the UDF object created in the system

Returns

BCM_E_NONE	UDF get successful.
BCM_E_NOT_FOUND	UDF does not exist.
BCM_E_UNIT	Invalid BCM Unit number.
BCM_E_INIT	UDF module not initialized.
BCM_E_UNAVAIL	Feature not supported.
BCM_E_XXX	Standard error code.

bcm_udf_get_all

Fetches all existing UDF ids

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_get_all(
    int unit,
    int max,
    bcm_udf_id_t *udf_id_list,
    int *actual);
```

Parameters

unit	(IN) Unit number.
max	(IN) Max number of UDF IDs
udf_id_list	(OUT) List of UDF IDs
actual	(OUT) Actual udfs retrieved

Description

Fetches all existing UDF ids

Returns

BCM_E_NONE	UDF get successful.
BCM_E_UNIT	Invalid BCM Unit number.
BCM_E_INIT	UDF module not initialized.
BCM_E_UNAVAIL	Feature not supported.
BCM_E_XXX	Standard error code.

bcm_udf_destroy

Destroys the UDF object

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_destroy(
    int unit,
    bcm_udf_id_t udf_id);
```

Parameters

`unit`	(IN) Unit number.
`udf_id`	(IN) UDF Object ID

Description

Destroys the UDF object

Returns

`BCM_E_NONE`	UDF deleted successfully.
`BCM_E_NOT_FOUND`	UDF does not exist.
`BCM_E_UNIT`	Invalid BCM Unit number.
`BCM_E_INIT`	UDF module not initialized.
`BCM_E_UNAVAIL`	Feature not supported.
`BCM_E_XXX`	Standard error code.

bcm_udf_pkt_format_create

Create a packet format entry

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_pkt_format_create(
    int unit,
    bcm_udf_pkt_format_options_t options,
    bcm_udf_pkt_format_info_t *pkt_format,
    bcm_udf_pkt_format_id_t *pkt_format_id);
```

Parameters

`unit`	(IN) Unit number.
`options`	(IN) API Options.
`pkt_format`	(IN) UDF packet format info structure
`pkt_format_id`	(IN/OUT) Packet format ID

Description

Create a packet format entry in the software. The entry will be installed only when the packet format entry is added to the udf.

Returns

`BCM_E_NONE`	UDF packet format entry created successfully.
`BCM_E_UNIT`	Invalid BCM Unit number.
`BCM_E_INIT`	UDF module not initialized.
`BCM_E_UNAVAIL`	Feature not supported.
`BCM_E_XXX`	Standard error code.

bcm_udf_pkt_format_info_get

Retrieve packet format info given the packet format Id

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_pkt_format_info_get(
    int unit,
    bcm_udf_pkt_format_id_t pkt_format_id,
    bcm_udf_pkt_format_info_t *pkt_format);
```

Parameters

`unit`	(IN) Unit number.
`pkt_format_id`	(IN) Packet format ID
`pkt_format`	(OUT) UDF packet format info structure

Description

Retrieve packet format info given the packet format Id

Returns

`BCM_E_NONE`	Packet format get successful.
`BCM_E_NOT_FOUND`	Packet format entry does not exist.
`BCM_E_UNIT`	Invalid BCM Unit number.
`BCM_E_INIT`	UDF module not initialized.
`BCM_E_UNAVAIL`	Feature not supported.
`BCM_E_XXX`	Standard error code.

bcm_udf_pkt_format_destroy

Destroy existing packet format entry

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_pkt_format_destroy(
    int unit,
    bcm_udf_pkt_format_id_t pkt_format_id);
```

Parameters

`unit`	(IN) Unit number.
`pkt_format_id`	(IN) Packet format ID

Description

Destroy existing packet format entry

Returns

`BCM_E_NONE`	Destroy packet format entry.
`BCM_E_NOT_FOUND`	Packet format ID does not exist.
`BCM_E_UNIT`	Invalid BCM Unit number.
`BCM_E_INIT`	UDF module not initialized.
`BCM_E_UNAVAIL`	Feature not supported.
`BCM_E_XXX`	Standard error code.

bcm_udf_pkt_format_add

Adds packet format entry to UDF object

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_pkt_format_add(
    int unit,
    bcm_udf_id_t udf_id,
    bcm_udf_pkt_format_id_t pkt_format_id);
```

Parameters

unit	(IN) Unit number.
udf_id	(IN) UDF ID
pkt_format_id	(IN) Packet format ID

Description

Adds packet format entry to UDF object

Returns

BCM_E_NONE	Packet format entry added to UDF successfully.
BCM_E_NOT_FOUND	UDF Id or packet format entry does not exist.
BCM_E_UNIT	Invalid BCM Unit number.
BCM_E_INIT	UDF module not initialized.
BCM_E_UNAVAIL	Feature not supported.
BCM_E_XXX	Standard error code.

bcm_udf_pkt_format_get

Deletes packet format spec associated with the UDF

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_pkt_format_get(
    int unit,
    bcm_udf_pkt_format_id_t pkt_format_id,
    int max,
    bcm_udf_id_t *udf_id_list,
    int *actual);
```

Parameters

`unit`	(IN) Unit number.
`pkt_format_id`	(IN) UDF Packet format ID
`max`	(IN) Max number of UDF IDs
`udf_id_list`	(OUT) List of UDF IDs
`actual`	(OUT) Actual udfs retrieved

Description

Deletes packet format spec associated with the UDF

Returns

`BCM_E_NONE`	Success.
`BCM_E_NOT_FOUND`	Packet format not found.
`BCM_E_UNIT`	Invalid BCM Unit number.
`BCM_E_INIT`	UDF module not initialized.
`BCM_E_UNAVAIL`	Feature not supported.
`BCM_E_XXX`	Standard error code.

bcm_udf_pkt_format_delete

Deletes packet format spec associated with the UDF

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_pkt_format_delete(
    int unit,
    bcm_udf_id_t udf_id,
    bcm_udf_pkt_format_id_t pkt_format_id);
```

Parameters

`unit`	(IN) Unit number.
`udf_id`	(IN) UDF ID
`pkt_format_id`	(IN) Packet format ID

Description

Deletes packet format spec associated with the UDF

Returns

`BCM_E_NONE`	Packet format configuration successfully deleted from UDF.
`BCM_E_NOT_FOUND`	UDF Id or packet format entry does not exist.
`BCM_E_UNIT`	Invalid BCM Unit number.
`BCM_E_INIT`	UDF module not initialized.
`BCM_E_UNAVAIL`	Feature not supported.
`BCM_E_XXX`	Standard error code.

bcm_udf_pkt_format_get_all

Retrieves the user defined format specification configuration from UDF

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_pkt_format_get_all(
    int unit,
    bcm_udf_id_t udf_id,
    int max,
    bcm_udf_pkt_format_id_t *pkt_format_id_list,
    int *actual);
```

Parameters

`unit`	(IN) Unit number.
`udf_id`	(IN) UDF ID
`max`	(IN) Max Packet formats attached to a UDF object
`pkt_format_id_list`	(OUT) List of packet format entries added to the udf id
`actual`	(OUT) Actual number of Packet formats retrieved

Description

Retrieves the user defined format specification configuration from UDF

Returns

BCM_E_NONE	Success.
BCM_E_NOT_FOUND	Either the UDF or packet format entry does not exist.
BCM_E_UNIT	Invalid BCM Unit number.
BCM_E_INIT	UDF module not initialized.
BCM_E_UNAVAIL	Feature not supported.
BCM_E_XXX	Standard error code.

bcm_udf_pkt_format_delete_all

Deletes all packet format specs associated with the UDF

Syntax

```
#include <bcm/udf.h>
int
bcm_udf_pkt_format_delete_all(
    int unit,
    bcm_udf_id_t udf_id);
```

Parameters

unit	(IN) Unit number.
udf_id	(IN) UDF ID

Description

Deletes all packet format specs associated with the UDF

Returns

BCM_E_NONE	Deletes all packet formats associated with the UDF.
BCM_E_NOT_FOUND	UDF Id does not exist.
BCM_E_UNIT	Invalid BCM Unit number.
BCM_E_INIT	UDF module not initialized.
BCM_E_UNAVAIL	Feature not supported.
BCM_E_XXX	Standard error code.

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 80:

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 81:

	sdk-6.4.1	sdk-6.4.0	sdk-6.3.7	sdk-6.2.9
golden	154	154	154	154
warmboot	288	278	284	254
auth	17	N/A	17	N/A
bfd	37	16	37	16
bhh	15	15	15	15
chip	9	N/A	10	N/A
cint	55	N/A	55	52
coe	510	N/A	511	N/A
cosq	306	294	296	289
custom	7	N/A	7	N/A
dvapi*	N/A	1003	N/A	985
ea	108	N/A	108	108
eav	19	N/A	19	N/A
extender	12	N/A	12	N/A
fabric	7	N/A	7	N/A
failover	8	N/A	8	N/A
fcoe	37	33	37	N/A
field	734	711	719	704
higigproxy	129	129	129	N/A
infra	114	N/A	114	114
ipfix	17	N/A	17	N/A
ipmc	56	N/A	54	N/A
l2	230	229	230	222
l2gre	13	N/A	13	N/A
l3	241	219	239	211
l3.alpm.combined	63	63	63	N/A
l3.alpm.combined.64	51	51	51	N/A
l3.alpm.parallel	63	63	63	N/A
l3.alpm.parallel.64	51	43	51	N/A
link	26	N/A	26	N/A
mim	19	N/A	19	N/A
mirror	146	N/A	146	N/A
misc	16	N/A	14	N/A
mpls	128	81	81	80
multicast	17	N/A	17	N/A
niv	13	N/A	13	N/A
oam	133	N/A	N/A	N/A
pkt	44	N/A	44	N/A
port	261	N/A	260	N/A
proxy	23	N/A	23	N/A
ptp	115	115	115	1
qos	12	N/A	12	N/A

Table 81:

	sdk-6.4.1	sdk-6.4.0	sdk-6.3.7	sdk-6.2.9
rate	21	N/A	21	N/A
rtag7	24	13	22	6
rx	21	N/A	21	N/A
ser	52	52	52	N/A
stack	49	49	49	49
stat	203	98	200	65
stg	42	N/A	42	N/A
switch	130	N/A	127	N/A
time	16	N/A	16	N/A
tlvMsg	13	13	13	13
trill	40	40	40	36
trunk	173	173	173	139
tunnel	65	65	65	65
subport	33	12	12	12
vlan	207	199	199	186
vxlan	69	42	58	41
wlan	17	N/A	17	N/A
Test Suite Total	5479	4253	5209	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 82:

	sdk-6.4.1	*sdk-6.4.0*	*sdk-6.3.7*	*sdk-6.2.9*
golden	1.5 %	2.4 %	1.8 %	5.1 %
warmboot	1.5 %	3.2 %	1.5 %	8.2 %
bcm.auth	0.2 %	N/A	5.1 %	N/A
bcm.bfd	0.9 %	0.2 %	0.7 %	0.0 %
bcm.bhh	2.0 %	2.1 %	2.2 %	5.5 %
bcm.chip	1.7 %	N/A	10.8 %	N/A
bcm.cint	0.0 %	N/A	0.0 %	13.0 %
bcm.coe	0.4 %	N/A	0.1 %	N/A
bcm.cosq	1.9 %	1.7 %	1.9 %	3.2 %
bcm.custom	0.0 %	N/A	0.0 %	N/A
bcm.dvapi	N/A	2.6 %	N/A	3.4 %
bcm.ea	0.0 %	N/A	0.0 %	0.2 %
bcm.eav	0.0 %	N/A	0.0 %	N/A
bcm.extender	0.1 %	N/A	0.1 %	N/A
bcm.fabric	0.0 %	N/A	0.0 %	N/A
bcm.failover	0.0 %	N/A	0.0 %	N/A
bcm.fcoe	0.1 %	0.0 %	0.0 %	N/A
bcm.field	1.6 %	2.0 %	1.2 %	2.2 %
bcm.higigproxy	0.9 %	1.6 %	0.9 %	N/A
bcm.infra	0.0 %	N/A	0.0 %	0.0 %
bcm.ipfix	0.7 %	N/A	0.7 %	N/A
bcm.ipmc	0.8 %	N/A	0.5 %	N/A
bcm.l2	1.5 %	3.1 %	2.1 %	1.8 %
bcm.l2gre	0.0 %	N/A	0.0 %	N/A
bcm.l3	1.0 %	2.6 %	1.4 %	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	0.1 %	N/A	0.1 %	N/A
bcm.mim	0.1 %	N/A	0.0 %	N/A
bcm.mirror	2.8 %	N/A	3.0 %	N/A
bcm.misc	0.8 %	N/A	0.7 %	N/A
bcm.mpls	0.6 %	1.6 %	0.5 %	1.3 %
bcm.multicast	2.6 %	N/A	0.6 %	N/A
bcm.niv	0.1 %	N/A	0.0 %	N/A
bcm.oam	1.1 %	N/A	0.3 %	N/A
bcm.perf	N/A	N/A	N/A	N/A

Table 82:

	sdk-6.4.1	sdk-6.4.0	sdk-6.3.7	sdk-6.2.9
bcm.pkt	0.0 %	N/A	0.0 %	N/A
bcm.port	1.5 %	N/A	1.6 %	N/A
bcm.proxy	0.7 %	N/A	0.7 %	N/A
bcm.ptp	0.0 %	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	N/A	0.0 %	N/A
bcm.rate	0.9 %	N/A	0.9 %	N/A
bcm.rtag7	0.0 %	0.0 %	0.2 %	0.2 %
bcm.rx	0.9 %	N/A	0.3 %	N/A
bcm.ser	0.5 %	0.3 %	1.7 %	N/A
bcm.stack	0.2 %	0.1 %	0.3 %	1.0 %
bcm.stat	1.0 %	1.9 %	1.0 %	4.5 %
bcm.stg	0.3 %	N/A	0.2 %	N/A
bcm.switch	1.0 %	N/A	2.7 %	N/A
bcm.time	0.0 %	N/A	0.0 %	N/A
bcm.tlvMsg	0.0 %	N/A	0.0 %	0.0 %
bcm.trill	1.1 %	2.9 %	0.4 %	10.5 %
bcm.trunk	2.0 %	2.1 %	1.6 %	1.4 %
bcm.tunnel	0.0 %	0.3 %	0.0 %	0.2 %
bcm.subport	1.7 %	6.8 %	3.2 %	36.5 %
bcm.vlan	1.2 %	2.0 %	1.1 %	1.4 %
bcm.vxlan	0.1 %	0.0 %	0.0 %	2.2 %
bcm.wlan	1.8 %	N/A	1.1 %	N/A
Test Suite Total	1.1 %	1.9 %	1.2 %	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 83:

	sdk-6.4.1	sdk-6.4.0	sdk-6.3.7	sdk-6.2.9
golden	0.1 %	0.0 %	0.0 %	6.5 %
warmboot	1.2 %	2.3 %	1.2 %	11.3 %
bcm.auth	0.0 %	N/A	5.9 %	N/A
bcm.bfd	0.0 %	0.0 %	0.0 %	0.0 %
bcm.bhh	N/A	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	N/A	10.0 %	N/A
bcm.cint	0.0 %	N/A	0.0 %	13.5 %
bcm.coe	0.0 %	N/A	0.0 %	N/A
bcm.cosq	0.7 %	0.3 %	0.7 %	3.1 %
bcm.custom	0.0 %	N/A	0.0 %	N/A
bcm.dvapi	N/A	1.2 %	N/A	2.7 %
bcm.ea	0.0 %	N/A	0.0 %	0.0 %
bcm.eav	0.0 %	N/A	0.0 %	N/A
bcm.extender	0.0 %	N/A	0.0 %	N/A
bcm.fabric	0.0 %	N/A	0.0 %	N/A
bcm.failover	0.0 %	N/A	0.0 %	N/A

Table 83:

	sdk-6.4.1	***sdk-6.4.0***	***sdk-6.3.7***	***sdk-6.2.9***
bcm.fcoe	2.7 %	3.6 %	2.7 %	N/A
bcm.field	0.9 %	0.8 %	0.4 %	1.1 %
bcm.higigproxy	0.6 %	0.8 %	0.8 %	N/A
bcm.infra	0.0 %	N/A	0.0 %	0.0 %
bcm.ipfix	0.0 %	N/A	0.0 %	N/A
bcm.ipmc	3.1 %	N/A	0.0 %	N/A
bcm.l2	0.8 %	1.3 %	1.3 %	0.0 %
bcm.l2gre	0.0 %	N/A	0.0 %	N/A
bcm.l3	0.6 %	0.9 %	0.4 %	0.5 %
bcm.l3.alpm.combined	0.0 %	0.0 %	0.0 %	N/A
bcm.l3.alpm.combined.64	0.0 %	N/A	0.0 %	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 %	N/A	0.0 %	N/A
bcm.mim	0.0 %	N/A	0.0 %	N/A
bcm.mirror	0.0 %	N/A	0.0 %	N/A
bcm.misc	1.8 %	N/A	0.0 %	N/A
bcm.mpls	0.8 %	1.7 %	1.2 %	1.3 %
bcm.multicast	0.0 %	N/A	0.0 %	N/A
bcm.niv	0.0 %	N/A	0.0 %	N/A
bcm.oam	0.8 %	N/A	0.0 %	N/A
bcm.pkt	0.0 %	N/A	0.0 %	N/A
bcm.port	1.3 %	N/A	0.8 %	N/A
bcm.proxy	0.0 %	N/A	0.0 %	N/A
bcm.ptp	0.0 %	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	N/A	0.0 %	N/A
bcm.rate	0.0 %	N/A	0.0 %	N/A
bcm.rtag7	0.0 %	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	N/A	0.0 %	N/A
bcm.ser	3.8 %	1.9 %	3.8 %	N/A
bcm.stack	0.3 %	0.0 %	0.0 %	0.0 %
bcm.stat	0.5 %	0.0 %	N/A	3.1 %
bcm.stg	0.0 %	N/A	0.0 %	N/A
bcm.switch	1.1 %	N/A	4.7 %	N/A
bcm.time	0.0 %	N/A	0.0 %	N/A
bcm.tlvMsg	0.0 %	N/A	0.0 %	0.0 %
bcm.trill	0.0 %	5.0 %	0.0 %	25.0 %
bcm.trunk	2.0 %	2.3 %	2.9 %	1.4 %
bcm.tunnel	0.0 %	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %	50.0 %
bcm.vlan	0.0 %	0.0 %	0.5 %	1.1 %
bcm.vxlan	0.0 %	0.0 %	0.0 %	24.4 %
bcm.wlan	0.0 %	N/A	0.0 %	N/A
Test Suite Total	0.7 %	0.9 %	0.7 %	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 84:

	sdk-6.4.1	*sdk-6.4.0*	*sdk-6.3.7*	*sdk-6.2.9*
golden	0.2 %	0.0 %	1.0 %	8.9 %
warmboot	1.4 %	2.5 %	1.0 %	8.4 %
bcm.auth	0.0 %	N/A	5.3 %	N/A
bcm.bfd	0.0 %	0.0 %	0.0 %	0.0 %
bcm.bhh	2.2 %	0.0 %	5.8 %	0.0 %
bcm.chip	0.0 %	N/A	11.0 %	N/A
bcm.cint	0.0 %	0.0 %	0.0 %	13.5 %
bcm.coe	0.0 %	N/A	0.0 %	N/A
bcm.cosq	1.0 %	0.0 %	0.8 %	3.3 %
bcm.custom	0.0 %	N/A	0.0 %	N/A
bcm.dvapi	N/A	1.8 %	N/A	3.8 %
bcm.ea	0.0 %	N/A	0.0 %	0.0 %
bcm.eav	0.0 %	N/A	0.0 %	N/A
bcm.extender	0.0 %	N/A	0.9 %	N/A
bcm.fabric	0.0 %	N/A	0.0 %	N/A
bcm.failover	0.0 %	N/A	0.0 %	N/A
bcm.fcoe	0.0 %	0.0 %	0.0 %	N/A
bcm.field	3.6 %	8.0 %	3.5 %	4.6 %
bcm.higigproxy	0.8 %	2.3 %	0.8 %	N/A
bcm.infra	0.0 %	N/A	0.0 %	0.0 %
bcm.ipfix	0.0 %	N/A	0.6 %	N/A
bcm.ipmc	0.0 %	N/A	0.5 %	N/A
bcm.l2	1.3 %	2.6 %	2.6 %	2.3 %
bcm.l2gre	0.0 %	N/A	0.0 %	N/A
bcm.l3	0.4 %	2.3 %	0.8 %	4.8 %
bcm.link	0.0 %	N/A	0.0 %	N/A
bcm.mim	0.0 %	N/A	0.0 %	N/A
bcm.mirror	0.0 %	N/A	1.4 %	N/A
bcm.misc	2.3 %	N/A	2.1 %	N/A
bcm.mpls	1.4 %	1.7 %	0.0 %	0.3 %
bcm.multicast	5.9 %	N/A	0.0 %	N/A
bcm.niv	0.0 %	N/A	0.0 %	N/A
bcm.oam	0.8 %	N/A	0.0 %	N/A
bcm.pkt	0.0 %	N/A	0.0 %	N/A
bcm.port	2.4 %	N/A	3.3 %	N/A
bcm.proxy	0.5 %	N/A	0.9 %	N/A
bcm.ptp	0.0 %	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	N/A	0.0 %	N/A
bcm.rate	3.3 %	N/A	3.3 %	N/A
bcm.rtag7	0.0 %	0.0 %	0.0 %	0.0 %
bcm.rx	0.9 %	N/A	0.6 %	N/A
bcm.ser	0.0 %	0.0 %	0.0 %	N/A
bcm.stack	0.7 %	0.0 %	1.0 %	4.8 %
bcm.stat	0.5 %	0.0 %	2.5 %	17.6 %
bcm.stg	0.0 %	N/A	0.0 %	N/A
bcm.switch	1.1 %	N/A	3.7 %	N/A

Table 84:

	sdk-6.4.1	*sdk-6.4.0*	*sdk-6.3.7*	*sdk-6.2.9*
bcm.time	0.0 %	N/A	0.0 %	N/A
bcm.tlvMsg	0.0 %	N/A	0.0 %	0.0 %
bcm.trill	3.1 %	5.0 %	3.3 %	27.8 %
bcm.trunk	0.8 %	1.2 %	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 %	2.0 %	1.0 %	2.6 %
bcm.vxlan	0.0 %	0.0 %	0.0 %	0.0 %
bcm.wlan	3.2 %	N/A	3.5 %	N/A
Test Suite Total	1.5 %	2.5 %	1.7 %	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 85:

	sdk-6.4.1	*sdk-6.4.0*	*sdk-6.3.7*
golden	0.2 %	3.2 %	0.1 %
warmboot	2.1 %	4.1 %	0.4 %
bcm.auth	0.0 %	N/A	5.9 %
bcm.bfd	N/A	N/A	N/A
bcm.bhh	N/A	0.0 %	N/A
bcm.chip	0.0 %	N/A	21.4 %
bcm.cint	0.0 %	N/A	0.0 %
bcm.coe	0.0 %	N/A	0.0 %
bcm.cosq	0.0 %	0.3 %	0.3 %
bcm.custom	0.0 %	N/A	0.0 %
bcm.dvapi	N/A	1.4 %	N/A
bcm.ea	0.0 %	N/A	0.0 %
bcm.eav	0.0 %	N/A	0.0 %
bcm.extender	0.0 %	N/A	0.0 %
bcm.fabric	0.0 %	N/A	0.0 %
bcm.failover	0.0 %	N/A	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.7 %	1.7 %	0.8 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	N/A	0.0 %
bcm.ipfix	0.0 %	N/A	0.0 %
bcm.ipmc	0.0 %	N/A	0.0 %
bcm.l2	0.3 %	1.3 %	1.1 %
bcm.l2gre	0.0 %	N/A	0.0 %
bcm.l3	0.4 %	0.9 %	0.8 %
bcm.link	0.0 %	N/A	0.0 %
bcm.mim	0.0 %	N/A	0.0 %
bcm.mirror	0.0 %	N/A	0.0 %
bcm.misc	0.0 %	N/A	0.0 %

Table 85:

	sdk-6.4.1	sdk-6.4.0	sdk-6.3.7
bcm.mpls	0.0 %	0.9 %	0.0 %
bcm.multicast	0.0 %	N/A	0.0 %
bcm.niv	0.0 %	N/A	0.0 %
bcm.pkt	0.0 %	N/A	0.0 %
bcm.port	1.7 %	N/A	1.7 %
bcm.proxy	0.0 %	N/A	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	N/A	0.0 %
bcm.rate	0.0 %	N/A	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	N/A	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.1 %	0.0 %	0.1 %
bcm.stg	0.0 %	N/A	0.0 %
bcm.switch	1.5 %	N/A	2.4 %
bcm.time	0.0 %	N/A	0.0 %
bcm.tlvMsg	0.0 %	N/A	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.5 %	1.2 %	0.4 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	1.5 %	2.9 %	1.4 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	N/A	0.0 %
Test Suite Total	0.4 %	1.1 %	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.

HELIX4

Table 86:

	sdk-6.4.1	sdk-6.4.0	sdk-6.3.7
golden	0.3 %	1.3 %	0.0 %
warmboot	2.4 %	3.6 %	3.2 %
bcm.auth	0.0 %	N/A	5.9 %
bcm.bfd	N/A	0.0 %	N/A
bcm.bhh	N/A	0.0 %	N/A
bcm.cfd	N/A	N/A	7.1 %
bcm.chip	0.0 %	N/A	10.0 %
bcm.cint	0.0 %	N/A	1.8 %
bcm.coe	0.0 %	N/A	0.0 %
bcm.cosq	0.7 %	0.0 %	0.7 %
bcm.custom	0.0 %	N/A	0.0 %
bcm.dvapi	N/A	1.8 %	N/A
bcm.ea	0.0 %	N/A	0.0 %

Table 86:

	sdk-6.4.1	*sdk-6.4.0*	*sdk-6.3.7*
bcm.eav	0.0 %	N/A	0.0 %
bcm.extender	0.0 %	N/A	0.0 %
bcm.fabric	0.0 %	N/A	0.0 %
bcm.failover	0.0 %	N/A	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	1.5 %	1.1 %	1.0 %
bcm.higigproxy	0.8 %	2.3 %	0.8 %
bcm.infra	0.0 %	N/A	0.0 %
bcm.ipfix	0.0 %	N/A	0.0 %
bcm.ipmc	0.4 %	N/A	0.0 %
bcm.l2	0.4 %	3.9 %	1.7 %
bcm.l2gre	0.0 %	N/A	0.0 %
bcm.l3	0.1 %	2.7 %	0.8 %
bcm.link	0.0 %	N/A	0.0 %
bcm.mim	0.0 %	N/A	0.0 %
bcm.mirror	0.0 %	N/A	0.0 %
bcm.misc	0.0 %	N/A	0.0 %
bcm.mpls	0.0 %	0.9 %	0.0 %
bcm.multicast	0.0 %	N/A	0.0 %
bcm.niv	0.0 %	N/A	0.0 %
bcm.pkt	0.0 %	N/A	0.0 %
bcm.port	1.3 %	N/A	1.9 %
bcm.proxy	0.0 %	N/A	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	N/A	0.0 %
bcm.rate	4.8 %	N/A	4.8 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	N/A	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.5 %	1.0 %	0.5 %
bcm.stg	0.0 %	N/A	0.0 %
bcm.switch	0.0 %	N/A	1.6 %
bcm.time	0.0 %	N/A	0.0 %
bcm.tlvMsg	0.0 %	N/A	0.0 %
bcm.trill	0.0 %	5.0 %	2.5 %
bcm.trunk	0.6 %	0.6 %	0.6 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.6 %	2.0 %	2.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	N/A	0.0 %
Test Suite Total	0.6 %	1.4 %	0.8 %

*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.

KATANA2

Table 87:

	sdk-6.4.1	sdk-6.4.0	sdk-6.3.7
golden	1.1 %	2.6 %	1.2 %
warmboot	1.4 %	5.5 %	2.4 %
bcm.auth	0.0 %	N/A	5.9 %
bcm.bfd	0.0 %	N/A	0.0 %
bcm.bhh	5.4 %	0.0 %	6.7 %
bcm.chip	0.0 %	N/A	10.0 %
bcm.cint	0.0 %	N/A	0.0 %
bcm.coe	1.7 %	N/A	1.7 %
bcm.cosq	1.9 %	1.4 %	2.0 %
bcm.custom	0.0 %	N/A	0.0 %
bcm.dvapi	N/A	1.8 %	N/A
bcm.ea	0.0 %	N/A	0.0 %
bcm.eav	0.0 %	N/A	0.0 %
bcm.extender	0.0 %	N/A	0.0 %
bcm.fabric	0.0 %	N/A	0.0 %
bcm.failover	0.0 %	N/A	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.9 %	1.5 %	0.9 %
bcm.higigproxy	0.4 %	0.8 %	0.5 %
bcm.infra	0.0 %	N/A	0.0 %
bcm.ipfix	0.0 %	N/A	0.0 %
bcm.ipmc	0.0 %	N/A	0.0 %
bcm.l2	1.0 %	3.1 %	2.1 %
bcm.l2gre	0.0 %	N/A	0.0 %
bcm.l3	1.5 %	3.2 %	3.0 %
bcm.link	1.1 %	N/A	1.3 %
bcm.mim	0.0 %	N/A	0.0 %
bcm.mirror	0.0 %	N/A	0.0 %
bcm.misc	0.0 %	N/A	0.0 %
bcm.mpls	2.6 %	2.6 %	1.2 %
bcm.multicast	0.0 %	N/A	0.0 %
bcm.niv	0.0 %	N/A	0.0 %
bcm.pkt	0.0 %	N/A	0.0 %
bcm.port	2.8 %	N/A	2.9 %
bcm.proxy	0.0 %	N/A	0.0 %
bcm.ptp	0.0 %	N/A	0.0 %
bcm.qos	0.0 %	N/A	0.0 %
bcm.rate	0.0 %	N/A	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	N/A	0.0 %
bcm.ser	2.9 %	0.0 %	11.5 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.5 %	0.0 %	0.3 %
bcm.stg	0.3 %	N/A	0.0 %
bcm.switch	0.0 %	N/A	1.6 %
bcm.time	0.0 %	N/A	0.0 %

Table 87:

	sdk-6.4.1	sdk-6.4.0	sdk-6.3.7
bcm.tlvMsg	0.0 %	N/A	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.4 %	1.2 %	0.4 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	12.1 %	0.0 %	0.0 %
bcm.vlan	0.3 %	2.9 %	2.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	N/A	0.0 %
Test Suite Total	1.6 %	1.5 %	1.3 %

*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.

STATIC CODE QUALITY ANALYSIS

Continued progress in whittling down static analysis defects per plan.

Table 88:

	Initial Reported Issues	*Reported Issues SDK 6.3.3*	*Reported Issues SDK 6.3.4*	*Reported Issues SDK 6.3.5*	*Reported Issues SDK 6.3.6*	*Reported Issues SDK 6.3.7*	*Reported Issues SDK 6.4.1*
DNX	664	688	628	125	125	129	131
XGS	271	292	327	155	135	131	123
SBX	600	421	323	281	140	0	0
SerDes	147	147	133	101	91	76	74
Common	2827	408	188	360	244	245	200
Total	4509	1956	1599	1022	735	571	528

Section 5: Resolved Issues for 6.4.1

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

Table 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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	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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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.
SDK-54619		`88650_B0`	Added diag "cosq flush" to flush all egress queues per port

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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
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

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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().

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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().`
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".

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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"`.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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-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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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-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.
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	Adjusted cint_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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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 89:

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 89:

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		`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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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 89:

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-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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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-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 efficent 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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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`	All the SER's are enabled at initialization sequence. There is no need to enable them by script anymore.
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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-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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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 queing is enabled then we program cos_mode with value 0 in RQE_PORT_CONFIG .

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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 charactors
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

Table 89:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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-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.
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 would not be restored properly in katana/katana2 after warmboot .As part of fix During warmboot we scan packet based wred table and update the software profile.
SDK-57038		88650_B1	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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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).

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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 doesnot 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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
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.
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.

Table 89:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57278	775407	88650_A0 88650ACP_A0 88650_B0 88650_B1 88660_A0	QOS: There are total 16 Egress PCP VLAN profile for Arad. Error is invoked when allocating Egress PCP VLAN profile 15. Changed the number of Egress PCP VLAN profile can be used from 15 to 16.
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.
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 backgroud 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.

Table 89:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
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.
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.

Table 89:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
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.
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);

Table 89:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.1***
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
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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
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	Dirty bit is not set in case of remote link failure. So, the resulting state may not be in sync with the scache in cases of auto sync. Change is to set the dirty bit in case of link update. Also, link_mask2 is not being recovered properly after warmboot. Now, this will be updated to valid up ports.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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-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	Field groups auto expansion is not recovered in Level 2 Warm Boot. Allocated 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	Changes to interrupt handling as implemented in Diag shell reference application: - Unmasked port interrupts by default - Fixed some tables to be designated as dynamic and not configuration. Changes to reference application, no changes to the driver and default behavior.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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 'l2x_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 'l2x_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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
SDK-57889	789898	56340_A0	Aging thread runs independently , wait on semphore for age time and wakes up & runs the bulk operaion. 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.
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 ov
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 89:

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 initilized 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`	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 fails with invalid parameter. The issue is 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 89:

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.

Table 89:

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 89:

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-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 does not work as expected
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_?ndpoint_action_set() with the destination set as a trap, function may fail to properly update the classifier for up MEPs.

Table 89:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
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-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.
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 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-58398	793706	88750_A0	All the SER monitor registers set by default in order to enable interrupt assertion upon memory fault. When interrupt asserted, SW can log and perform a corrective action if needed.
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 .

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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 causeed 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.
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.

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.
SDK-58493		88650_A0	Masking all interrupt are not done for all instances of interrupt register. This is fixed now. Memories shadows did not updated by HW memory values after initialization sequence. This is also fixed.
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.

Table 89:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.1*
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.
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-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.
SDK-58824	798782	88650_B1	fixed incorrect behaviour of "memory-arrays" on cache.
SDK-58863	797630	88650_A0 88660_A0 88670_A0	arp extension feature: Backward compatibilty 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.
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

Table 89:

Number	CSP #	Chips	Release Notes For 6.4.1
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.

Section 6: Unresolved Issues for 6.4.1

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

Table 90:

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-42259		56440_A0 56440_A1	Spurious error messages may be seen when executing Rx/TX tests TR90 and TR91 when the KNET modules is loaded.
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, 1/* 10 */, -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, -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, 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 90:

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-49910		56640_A0 56640_A1 56640_B0	If the default schedule hierarchy is changed, subsequent calls to various "cos" commands may fail with an "INVALID PORT" error, "cos config" is one example.
SDK-51099	695521	88650_A0 88650_B0 88650_B1	In L2, in distributed systems under extreme scenario, the MAC Table reply FIFO may be empty but its interrupt is up. In this case, the interrupt should be reset before trying inserting a MAC Table address, otherwise a failure in the insertion will be returned by the Driver.
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 91: 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 91: 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	BCM53401 A0	24-port GbE plus 4-port 10GbE Multilayer Ethernet Switch with HiGi Uplinks
	BCM53402 A0	8-port 10GbE Multilayer Ethernet Switch

Table 91: Switch Devices

Family	Devices	Description
	BCM53405 A0	16-port 10GbE Multilayer Ethernet Switch
	BCM53406 A0	12-port 10GbE plus 8-port 2.5GbE and 4-port 5GbE/2.5GbE Multilayer Ethernet Switch
	BCM53408 A0	24-port 1GbE/2.5GbE Multilayer Ethernet Switch
	BCM53411 A0	24-port GbE plus 4-port 10GbE Multilayer Ethernet Switch with HiGi Uplinks and integrated CPU
	BCM53412 A0	8-port 10GbE Multilayer Ethernet Switch with integrated CPU
	BCM53415 A0	16-port 10GbE Multilayer Ethernet Switch with integrated CPU
	BCM53416 A0	12-port 10GbE plus 8-port 2.5GbE and 4-port 5GbE/2.5GbE Ethernet Switch with integrated CPU
	BCM53418 A0	24-port 1GbE/2.5GbE Multilayer 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

Table 91: Switch Devices

Family	Devices	Description
	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	
	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

Table 91: Switch Devices

Family	Devices	Description
	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
	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
	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	

Table 91: Switch Devices

Family	*Devices*	*Description*
	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+
	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	

Table 91: Switch Devices

Family	Devices	Description
	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	
	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

Table 91: Switch Devices

Family	Devices	Description
	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	BCM56040 A0	1xF.QSGMII + 3xF.HG[42] + 1GE
	BCM56041	Ranger device, meant for embedded connectivity supports 1Ge (port 49), 2 X GE (iPROC), Flex 4x10G, 3 X 4 X 10G
	BCM56042	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	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	BCM56450 A0	24-port GbE Multilayer Switch with 4-port 10 GbE uplinks, stacking, integrated CPU and Traffic Manager
	BCM56455	2 x 20GE (G.INT) + 2 x HG13
	BCM56456	1 x XAUI + 8 x GE
BCM56500	BCM56500 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56500 A1	

Table 91: 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 91: 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 91: 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 91: 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 91: 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 91: 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	
BCM56850	BCM56850 A1	1.28Tbps I/O, 1Tbps Core Ethernet Switch
	BCM56852 A2	100x10G, 960Gbps Multilayer Switch
	BCM56854 A1	
	BCM56751P A2	1.28Tbps I/O, 960Gbps Core Ethernet Switch Fabric
	BCM56751P A1	1.28Tbps I/O, 960Gbps Core Ethernet Switch Fabric
	BCM56830 A1	960Gbps Ethernet Switch
	BCM56830 A2	960Gbps Ethernet Switch
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 Queueing Engine with 49 SPI 4.2 subports
	QE-2000 A2	
	QE-2000 A3	
	QE-2000 A4	
BCM88230	BCM88230 A0	XGS Core (XCore/SBX) Fabric Queueing Engine with Integrated Traffic Management with 4 HiGig2 ports, 50Gbps

Table 91: Switch Devices

Family	Devices	Description
	BCM88230 B0	
	BCM88235 A0	XGS Core (XCore/SBX) Fabric Queueing 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 92: 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 93: Switch Devices that support Warm boot

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

Table 93: Switch Devices that support Warm boot

Family	Devices	Description
	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
	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

Table 93: Switch Devices that support Warm boot

Family	Devices	Description
	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	
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
	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	

Table 93: Switch Devices that support Warm boot

Family	Devices	Description
	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	
	BCM56508 B1	
	BCM56508 B2	
	BCM56509 A0	24-Port GbE Layer 2 Switch with Four 10-GbE/HiGig+ Ports
	BCM56509 A1	
	BCM56509 B0	

Table 93: Switch Devices that support Warm boot

Family	Devices	Description
	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	
	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	

Table 93: Switch Devices that support Warm boot

Family	Devices	Description
	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	
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	

Table 93: Switch Devices that support Warm boot

Family	Devices	Description
	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
	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	

Table 93: Switch Devices that support Warm boot

Family	Devices	Description
	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
	BCM88750 B0	

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

PHYS

Table 94: 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 Preview)
BCM54292_A0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (1588 feature is Preview)
BCM54294_A0	54280	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (1588 feature is Preview)
BCM54340_B0	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54340_C0	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)

Table 94: PHYs

Device	Driver Family	Description
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.
BCM8074_A0	8072	Quad-Channel Serial 10-GbE BASE-KR to XAUI Transceiver. Firmware version 010C.

Table 94: PHYs

Device	Driver Family	Description
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. Preview)
BCM8728_A0	8706	Dual-Channel 10-GbE SFI-to-XAUI(TM) Transceiver with EDC. Firmware version 0511. (Preview)
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 "(Preview)
BCM82328_B0	82328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version B "(Preview)
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 R026
BCM84333_B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.69 (Preview) (Needs additional software component)
BCM84334_B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.69 (Preview) (Needs additional software component)
BCM84336_B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.69 (Preview) (Needs additional software component)
BCM84793_A0	84793	100GbE/OTN 4x25/28G VSR28 to 10x10/11G CAUI Gearbox PHY. Firmware version 0xD009 (Preview - 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 (Preview)
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 preview)
BCM84834_B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.69(Driver support for IEEE 1588 features are preview)

Table 94: PHYs

Device	Driver Family	Description
BCM84836_B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.69(Driver support for IEEE 1588 features are preview)
BCM84844_A0	8481	Quad 10GBASE-T Transceiver. Firmware version 1.06(Driver support is preview)
BCM84846_A0	8481	Dual 10GBASE-T Transceiver. Firmware version 1.06(Driver support is preview)
BCM84848_A0	8481	Quad 10GBASE-T Transceiver. Firmware version 1.06(Driver support is preview)
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 preview)
BCM84729_A0	84729	Dual-Channel SFI to XAUI with Macsec, 1588 (Firmware version 0x124. Driver support for IEEE 1588 features are preview)
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 [Preview]
BCM84747_A0	84728	Quad SFI to XAUI with 1588 (Firmware version 0x124. Preview)
BCM84748_A0	84728	Quad SFI to XAUI with WAN/1588 (Firmware version 0x124. Preview)
BCM84749_A0	84749	Quad SFI to XAUI with Macsec, 1588 (Firmware version 0x124. Driver support for IEEE 1588 features are preview)
BCM84752 A0	84740	Dual-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version D105. (Preview)
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) [Preview]
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. (Preview)
BCM84780_A0	84740	Octal-Channel 10 GbE SFI-to-XFI PHY with 1588. Firmware version 0x128 (Preview)
BCM84784_A0	84740	Dual 40GbE/Octal 10GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version 0x125 (Preview)
BCM84764_A0	84728	Quad SFI to RXAUI with 1588 (Firmware version 0x124. Preview)
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. Preview)

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-PG817-R) for instructions on porting the SDK to another platform.

Table 95: 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 96: 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)
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 97: 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 98:

SDK	Firmware 2.0	Firmware 2.1	Firmware 2.2	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
SDK-6.2.0	Yes	No	No	No	No	No	No	No	No	No	No
SDK-6.2.1	No	No	Yes	No	No	No	No	No	No	No	No
SDK-6.2.3	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No
SDK-6.2.4	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No
SDK-6.2.5	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No
SDK-6.2.6	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No
SDK-6.2.7	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No
SDK-6.2.8	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No
SDK-6.2.9	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No
SDK-6.3.0	No	No	No	Yes	Yes	No	Yes	No	No	No	No
SDK-6.3.1	No	No	No	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.2	No	No	No	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.3	No	No	No	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.4	No	No	No	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.5	No	No	No	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.6	No	No	No	No	No	No	Yes	Yes	Yes	Yes	No
SDK-6.3.7	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes
SDK-6.4.0	No	No	No	No	No	No	Yes	Yes	Yes	No	No
SDK-6.4.1	No	No	No	No	No	No	No	No	Yes	Yes	Yes

BCM56640 FIRMWARE COMPATIBILITY MATRIX

Table 99:

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*
SDK-6.2.3	No	No	No	No	No	No	No	No
SDK-6.2.4	No	No	No	No	No	No	No	No
SDK-6.2.5	Yes	Yes	Yes	No	No	No	No	No
SDK-6.2.6	No	No	Yes	No	No	No	No	No
SDK-6.2.7	No	No	Yes	No	No	No	No	No
SDK-6.2.8	No	No	Yes	No	No	No	No	No
SDK-6.2.9	No	No	Yes	No	No	No	No	No
SDK-6.3.0	Yes	Yes	No	No	No	No	No	No
SDK-6.3.1	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.2	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.3	No	No	No	Yes*	Yes*	Yes*	No	No
SDK-6.3.4	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.5	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.6	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.7	No	No	No	Yes	Yes	Yes	No	Yes
SDK-6.4.0	No	No	No	Yes	Yes	Yes	No	No
SDK-6.4.1	No	No	No	No	No	Yes	Yes	Yes

BCM88650 FIRMWARE COMPATIBILITY MATRIX

Table 100:

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***
SDK-6.2.3	Yes	Yes	No	No	No	No	No	No
SDK-6.2.4	Yes	Yes	No	No	No	No	No	No
SDK-6.2.5	Yes	Yes	No	No	No	No	No	No
SDK-6.2.6	No	No	Yes	No	No	No	No	No
SDK-6.2.7	No	No	Yes	No	No	No	No	No
SDK-6.2.8	No	No	Yes	No	No	No	No	No
SDK-6.2.9	No	No	Yes	No	No	No	No	No
SDK-6.3.0	Yes	Yes	No	Yes	No	No	No	No
SDK-6.3.1	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.2	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.3	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.4	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.5	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.6	No	No	No	Yes	Yes	Yes	No	No
SDK-6.3.7	No	No	No	Yes	Yes	Yes	No	Yes
SDK-6.4.0	No	No	No	Yes	Yes	Yes	No	No
SDK-6.4.1	No	No	No	No	No	Yes	Yes	Yes

BCM56850 FIRMWARE COMPATIBILITY MATRIX

Table 101:

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***
SDK-6.2.6	Yes	No	No	No	No	No
SDK-6.2.7	Yes	No	No	No	No	No
SDK-6.2.8	No	Yes	No	No	No	No
SDK-6.2.9	No	Yes	No	No	No	No
SDK-6.3.0	No	Yes	No	No	No	No
SDK-6.3.1	No	Yes	Yes	Yes	No	No
SDK-6.3.2	No	Yes	Yes	Yes	No	No
SDK-6.3.3	No	Yes	Yes	Yes	No	No
SDK-6.3.4	No	Yes	Yes	Yes	No	No
SDK-6.3.5	No	Yes	Yes	Yes	No	No
SDK-6.3.6	No	Yes	Yes	Yes	Yes	No
SDK-6.3.7	No	Yes	Yes	Yes	Yes	Yes
SDK-6.4.0	No	Yes	Yes	Yes	No	No
SDK-6.4.1	No	No	No	Yes	Yes	Yes

BCM88030 FIRMWARE COMPATIBILITY MATRIX

Table 102:

SDK	*Firmware 3.2.0*	*Firmware 3.2.1*	*Firmware 3.2.2*	*Firmware 4.0.0*	*Firmware 4.0.1*
SDK-6.2.8	Yes	No	No	No	No
SDK-6.2.9	Yes	No	No	No	No
SDK-6.3.1	Yes	Yes	Yes	No	No
SDK-6.3.2	Yes	Yes	Yes	No	No
SDK-6.3.3	Yes	Yes	Yes	No	No
SDK-6.3.4	Yes	Yes	Yes	No	No
SDK-6.3.5	Yes	Yes	Yes	No	No
SDK-6.3.6	Yes	Yes	Yes	Yes	No
SDK-6.3.7	Yes	Yes	Yes	Yes	Yes
SDK-6.4.0	Yes	Yes	Yes	No	No
SDK-6.4.1	No	No	Yes	Yes	Yes

BCM56450 FIRMWARE COMPATIBILITY MATRIX

Table 103:

SDK	Firmware 2.0	Firmware 2.1	Firmware 2.2	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
SDK-6.3.6	Yes	No	No	No	No	No	No	No	No	Yes	No
SDK-6.3.7	Yes	No	No	No	No	No	No	No	No	Yes	Yes
SDK-6.4.0	No	No	No	No	No	No	No	No	No	No	No
SDK-6.4.1	No	No	No	No	No	No	No	No	Yes	Yes	Yes

BMACSEC SDK COMPATIBILITY MATRIX

Table 104:

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

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 105: 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 356)
ED Editor	USENET comp.sources.misc Volume 9, Issue 36	src/appl/diag/edline.c	See (ED Editor License terms and conditions) (page 358)
CINT	http://www.gnu.org/software/bison/	src/appl/cint/cint_parser.[ch]	See (CINT parser license terms and conditions) (page 359)
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 360)
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 361)
APIMODE	http://www.gnu.org/software/bison/	src/appl/diag/api/api_grammar.tab.[ch]	See (APIMODE parser license terms and conditions) (page 362)
VxWorks	Wind River Systems, Inc.	systems/vxworks	See (Wind River Systems license terms and conditions) (page 363)

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 '^' 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 359) 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 6.4.0

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

Table 106:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.0*
SDK-39298		56640_A0 56640_A1	Added a new feature to control metering in egress mode
SDK-39435	625583	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	Add supports for the dual-lane forced speed mode running with CL72. It requires FW version A041_003 or above.
SDK-41137	549821	All	On multicast removal all the ports are iterated to clear the ports associated with the multicast address. This iteration is not required as iteration on the port members of the multicast address will save more cylces of iteration. Fixed the Iteration based on the member ports of the multicast group.
SDK-41495	557384	56640_B0	Support has been added for Triumph 3 in bcm_switch_pkt_info_hash_get() API and incomplete functionality in compute load balancing and compute trunk hash has been corrected.
SDK-42899	558213	56640_A0 56850_A0	When the traffic is running, issuing a stats clear can cause the MMU unicast drop counters to get into a state, where the counts are not updated. The entry has 3 fields packet count, byte_count and parity field and all the fields need to be cleared if parity is not enabled. Fixed the issue. .
SDK-43520		56640_A0 56440_A0 56450_A0	When SDK is initialized along with Firmware, queue configuration is required to indicate the Rx queues mapped to external CPU (Ehost) and microcontrollers (UC0, UC1). If the queue configuration is not present then SDK init fails. An error message is now displayed indicating if SDK init fails due to to missing queue configuration.
SDK-44138	717410	56634_A0	snmpDot1dBasePortMtuExceededDiscards only counts packets dropped on Rx and does NOT include the count of packets dropped by the pipeline because they exceeded the configured max frame size. This counter will be equal to or less than the number of packets dropped due to MTU violations
SDK-44342	601905	56640_A0 56640_A1 56640_B0	Triumph3 chip supports attaching up to 3 counters for an entry in IFP. Field STAT APIs have been enhanced to support the same in SDK.
SDK-45115		88650_B0	BCM shell: Typing "diag pp MODE_info_Set ?" in BCM shell would cause segmentation fault.

Table 106:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.0*
SDK-46431		56334_B0	In earlier releases IP multicast configuration did not work on Enduro (BCM56334_B0). Made code changes to configure L3_IIFf in VLAN_TABm during init to allow this to work correctly.
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-46612	628861	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	Triumph 3 MAC Table Deletion Callbacks were are inconsistent in Polling mode . this has been fixed by the following: 1. The 'l2_entry_data_t->enable' was wrongly being set for FIFO mode where L2 data store is not needed, it is unset for FIFO, but set for POLL mode, this is done in 'soc_tr3_l2x_start' 2. On a MAC re-learn case, with callback's suppressed, the deleted entry was being wrongly sent to the callback handler (soc_l2_entry_callback) from within '_soc_l2x_sync_bucket', this is now changed to NULL, so that no-DEL callbacks are called in such cases
SDK-46734	636372	56448_B0 All 56440_A0 56445_A0 56440_A1 56445_A1 56444_A1 56450_A0 56449_B0 56445_B0 56440_B0 56447_B0 56443_B0 56441_B0 56446_B0	Introduced new flag 'BCM_PORT_MATCH_PORT_VLAN16' in bcm_port_match_t to match mod-port/ trunk+16 bit outer VLAN TAG for VLAN translation
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.

Table 106:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.0***
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-48018	652215	`56840_A0`	In earlier releases, `bcm_cosq_gport_bandwidth_get()` did not work. This has been resolved.
SDK-48130	663340	`56640_A0 56640_A1`	Added code to initialize rtag7 flow based hash related parameters to enable macroflow offset APIs.
SDK-48272	665127	`56334_B0 56334_A0`	In earlier releases `bcm_esw_mirror_port_get()` did not return correct flags. This has been resolved by removing the " if" condition which was preventing the proper update of the flag for the egress mirroring
SDK-48433	627988	`56224_B0 56224_A0`	Removed the check in SDK which prevents the customer from configuring both `BCM_L2_LEARN_LIMIT_ACTION_DROP` and `BCM_L2_LEARN_LIMIT_ACTION_CPU` at the same time when making calls to `bcm_l2_learn_limit_set()` The corresponding actions are supported in the hardware for RAPTOR/RAVEN/HAWKEYE devices.
SDK-48449		`56850_A2`	The support for one-lane port running CL72 is added for the JIRA. Customers need to put 1) `port_init_cl72=0x1` in their configuration file, or 2) call API to enable the CL72 mode, then set the port speed. This JIRA requires FW version `A041_002` or above.
SDK-49335		`56640_A0 56640_A1`	In an earlier release the TR3 shadow tables were not getting initialized correctly in some cases. In this release we are now initializing the arlShadow tables to correct values, applicable for TR3 when no external TCAM is in use. In `'soc_tr3_l2_attach', L2_ENTRY_1m, EXT_L2_ENTRY_1m` and `EXT_L2_ENTRY_2m` are initialized to their `'soc_mem_index_count's,` Previously they were being set 0 earlier.
SDK-49347		NA	Updated the grog file for `bcm_port_encap_config_t` documentation.
SDK-49464	681536	`88650_A0 88650_B0 88650_B1`	When the packet is trapped and parsed in the CPU, the Source-System-Port parsing was not considering the LAG case. This is fixed, by setting the `src_trunk` parameter for the LAG Id, and the `src_port` and `src_mod` parameters corresponding to the selected LAG member port.
SDK-49473	683076	`54680E_A1 54680E_B0 54682E_A1 54682E_B0 54685E_A1`	In earlier releases there were Display errors in the EEE command for BCM54685E. This has been fixed.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-49649		All	In an earlier release, in file src/soc/common/mem.c, `MEM_LOCK/MEM_UNLOCK` in functions `_soc_mem_read()` / `_soc_mem_write()` and `soc_mem_alpm_read()` / `soc_mem_alpm_wrte()` was done when doing S-Chan processing, but in functions `soc_mem_generic_insert()` / `soc_mem_generic_delete()` / `soc_mem_generic_lookup()` and `soc_mem_alpm_lookup()` / `soc_mem_alpm_insert()` / `soc_mem_alpm_delete()`, we were missing this, This protection has been added.
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-49746		`88650_A0 88650_B0 88650_B1`	Enable Bounce back filter for 2-pass trill: In the second pass going back to trill packets are filtered by bounce-back-filter.
SDK-49806		`88650_A0 88650_B0`	In PON application, in IPv6 Source bind implementation, the code has been changed to be more generic.
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-49829	684594	`56440_B0`	Fixed the code to configure the `RQE_PORT_CONFIGr` in the bcmPortControlCustomerQueuing switchcontrol set for Katana/Katana2
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-50029	682932	`56334_B0 56142_A0 56132_A0`	In earlier releases, FP qualifiers had a mismatch when there two FP groups on the multi slice. In this release we have modified the code such that we now install the selcodes in the slice if the slice is not empty and this is the first entry for the group in that slice.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
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-50066		`88660_A0`	In BCM88660, introduce new support for IPMC and IGMP after exiting tunnel (VXLAN, L2GRE, VPLS). See `cint_igmp_example.c` for application explanation and valid packet flows.
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-50121		`88650_A0`	KBP Serdes init sequence changed to use KBP SDK API. Internal implementation change, no affect on customer application
SDK-50142	690184	`56850_A0 56850_A1 56850_A2`	A request was received such that the hardware queue number could be retrieved when the system was transmiting packets from the cpu port to a front panel port in bypass mode. The new API is implemented to support the ability to retrieve hardware queue number in PBSMH header according to the port and cosq.
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-50148		`56850_A0 56850_A1 56850_A2`	Add BST software Snapshot improvement. This included adding the following capabilities. A) Enabled bst for all ingress and egress resources. B) Provided a sw trigger for taking snapshot . C) Allow the user to read the statistics without clearing the counters D) Added a separate api for clearing each specific resource counter. See API Section of this document. E) Added the ability to disable the snapshot mode via the api if needed. F) Enabled the ability to Disable BST G) 'BST "max used mode" and "current counting" mode are supported
SDK-50162	692128	All	Multiple RX interrupt packet handlers were being called when a packet was handled. This has been resolved.
SDK-50212		`56850_A2`	The code supports for 40G/42G HG FEC are added.
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-50288	692335	All	In ealier releases, BFD could not be supported in the LAN network because of IP addresses limitation. This has been fixed.
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

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
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
SDK-50377	686726	`56150_A0`	Add KNET support for switch devices attached via iProc AXI bus.
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-50481	692651	All	Modified to check fiber channel inner and outer fields with `BCM_FIELD_DATA_FORMAT_FIBRE_CHAN_ANY` instead of 0
SDK-50490	695720	`88030_A0`	Sync attribute has been added to CMU counter config. Valid values are "true" or "false".
SDK-50498	696599	All	Add KNET support for BCM56150 family of devices.
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.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-50595		88650_A0 88650_B0	ERSPAN ARAD: Outbound Multicast ERSPAN mirroring is not supported in default application. To support multicast ERSPAN outbound mirroring a new soc property introduce "custom_feature_erspan_mc_support=1". In case it is set , first 16 entries in ISID-table are used for ERSPAN feature. When custom feature is enabled, User can allocate for MIM, VXLAN, L2GRE only VPNs that pass the constraint (vsi & 0xFFF) > 16. Additionally, VPN must be allocated for those application WITH-ID only. See more details in cint_mirror_erspan.c
SDK-50611	684857	88650_A0	In Field Processor in Egress stage, an HW limitation requires that none or both lookup keys are valid in Egress PMF. If a single lookup key is valid, the second lookup key returned result will be invalid once used in the future. To handle this limitation, the Driver uses the last TCAM DB Profile (ID 47). It allows the user to define only 47 TCAM Databases instead of 48. To disable this implementation (e.g. if Egress Field Processor is not used), set the SOC property custom_feature_egress_pmf_lookups_always_valid_disable=1.
SDK-50637	689475	All	Added new soc property eb2_2bytes_big_endian to support EB2 endianism
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

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-50753		88650_A0 88650_B0 88650_B1 88660_A0	Add new diag "diag ing_congestion" to display ingress global resource.
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)

Table 106:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.0*
SDK-50757		88650_A0 88650_B0 88650_B1 88660_A0	Added diagnostics "diag rates sch" to display E2E scheduler rate.
SDK-50758		88650_A0 88650_B0 88650_B1 88660_A0	Added diagnostics "Gtimer" to control gtimer in sub-block for rate calculation. Added interval option for "diag counter".
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-50823	699173	88650_A0 88650_B0 88650_B1	At egress, the user can define a packet to be trapped and sent to the CPU. By default, the trap profile (action profile) was sending the packet to the Egress Queue Pair with ID = CPU Port number instead of sending to the CPU. This is fixed
SDK-50828	686923	88650_B1	STG: bcm_stg_vlan_add() and bcm_stg_stp_set() do not return BCM_E_NOT_FOUND when passed in a spanning tree group that does not exist. Update verification on those functions.
SDK-50836		88650_A0 88650_B0 88650_B1 88660_A0	Add new diag(tdm edit show [port=15]) to display tdm edit information.
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-50859	697873	56850_A0	It was discovered that 15 profiles could be created for the mapping from internal priority to MPLS Exp at the egress but 16 profiles could not be created successfully when repeating the profiles creation although the system allowed the user to attempt this . It is fixed in this release and max 15 profiles can be created.
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 106:

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-50982	703790	56850_A0	Add BCM shell CLI support and HG_TRUNK mode for packet hash select API.
SDK-51019	687800	56850_A1	Implemented in the new policer mode bcmPolicerGroupModeShortIntPri for creation of 8 internal policers.
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-51065	705285	All 56643_A0 56643_A1 56643_B0	Updated to make triumph_3 devices boot in 64 port single modid mode by default and if config variable spn_MODULE_64PORTS = 0 then boot in dual modid mode
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-51127	702045	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 during qualifiers init.
SDK-51154	701733	56640_A0 56440_A0 56643_A0 56644_A0 56648_A0 56850_A0 56445_A0 56440_A1 56445_A1 56444_A1 56340_A0 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 56850_A1 56443_B0 56441_B0 56446_B0 56448_B0 56850_A2 56342_A0 56442_B0	Added bcmFieldActionETagNew [Add/Change ETAG] & bcmFieldActionETagDelete [Delete ETAG] in IFP to support Port Extenders Etag add/delete/change options.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
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-51230	708240	All	Support has been added for new API to retrieve member port for DLB HG Trunk
SDK-51340	709181	88650_B1	Upon FEC creation (bcm_l3_egress_create) Correct the verification of LAG ID to allow also group IDs that are higher than 32.
SDK-51348		88650_A0	To end a tdm session using you have two options: 1.Set destination port to an invalid destination - BCM_GPORT_BLACK_HOLE. 2.Call bcm_port_control_set () with type bcmPortControlRxEnable. Be advised, for tdm ports, when disabling a port rx, the valid range cell size min filter must be of 192B or above, and this configuration affects all of the other tdm sessions.
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-51405		56640_A0 56640_A1 56640_B0	L2 entries learnt on the trunk ports were not being deleted on ring flush. The trunk ports are now matched with the Trunk port module ID (0x80) and the entry is deleted on ring flush in addition to the line ports.
SDK-51451		56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56850_A0 56843_B0 56841_A3 56846_A1 56841_B0 56850_A1 56850_A2	Support for BCM_L2_STATION_COPY_TO_CPU configuration in l2 station entry for TD/TD2/TT2 has been added.
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.

Table 106:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.0*
SDK-51506		56640_A0 56548_A0 56546_A0 56545_A0 56544_A0 56542_A0 56541_A0 56540_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 56545_A1 56540_B0 56541_B0 56546_B0 56544_B0 56547_A0 56545_B0 56542_B0	Fixed issue in handling flushing MAC entries by Virtual Port's correctly. On Triumph3, the `key_type` was not being set correctly for Flush-by-VP calls. The `key_type` for MPLS, MiM, L2GRE, VXLAN VFI types needs to be set to VFI type
SDK-51521		88650_B1	Diagnostics: "diag pp `Parsing_Info"` sometimes output incorrect `inner_vid` value due to wrong initialization.
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.
SDK-51553	710528	56850_A0 56850_A1 56850_A2	Corrected the `VLAN_PROFILEm` configuration flow in qos module.
SDK-51568	705719	56850_A0 56850_A1 56830_A1 56850_A2 56830_A0 56830_A2	In earlier releases `BCM56850_SVK` would automatically reboot within 1 minute if `"table_dma_enable=0"` is set in config.bcm or in SVK flash. This has been fixed by modifying memory scan with dma-disabled.
SDK-51597	704238	56224_B0 56224_A0	Made code changes to allow '-1' as valid port parameter in `bcm_vlan_translate_add()` for 56224 devices.'-1' is valid for 56224 as given in the Programmer's reference guide which indicates configuration on all the ports. The function `bcm_esw_vlan_translate_add()` should now able to configure for '-1' as well for 56224 devices.
SDK-51599	712774	All	In earlier releases SDK cli "mc show" did not display all OIFs for a multicast groupfixed to support any number of OIFs. This has been fixed to support any number of OIFs.
SDK-51643		56340_A0	Fixed and tested on GTO/BCM56340A. BCM init and rc failures aren't happening anymore.
SDK-51645	713523	56850_A0	Renamed conflicting #define `MAX_NAME_LEN` to `SOC_MAX_NAME_LEN`

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-51652	703012	88650_A0 88650_B0 88660_A0	L3 Egress object optimization: When connecting between MPLS/IP tunnel to link-layer (bcm_l3_egress_create), API always set Link-layer information even in case no Link-layer information has been modified. Use combination of flags BCM_L3_KEEP_DSTMAC, BCM_L3_KEEP_VLAN and BCM_L3_REPLACE and valid encap_id to modify only connection between MPLS/IP tunnel to link-layer. No link-layer modifications are done.
SDK-51665	713519	All	Add vlan control vlan selective set/get API to control UMC_IDXf/BC_IDXf/UUC_IDXf in Vlan Table
SDK-51677	695953	88650_B0 88650_B1	When egress packets are dropped at the EGQ, i.e. EGQ-delete-queue is receiving traffic, it'll take priority over the NIF ports, and might cause packet drop. After the fix delete queue will get priority over NIF only if it is almost full.
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-51811	713635	56640_A0 56850_A0 56640_A1 56640_B0 56850_A1 56850_A2	Support for symmetric hash for Resilient Hashing has been added for TD2/TR3.
SDK-51814		88650_A0 88660_A0	When configuring VRRP for ARAD+, up to 4k VSIs can be assigned to each VRID. There was an error that made it impossible to delete a VRID if exactly 4k (4096) VSIs entries were assigned to it. The error is now fixed.
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-51823	716406	88030_A0	ppe config error carse variable length of packet header not work. modify ppe associate structure define to fix this issue.
SDK-51824	716986	All	Added a new command 'ser inject' which may be used to inject single-bit parity errors into memories while having a minimal effect on system state.
SDK-51827	716807	88030_A0	Resolves ucode reload issue seen in 2_146 and TOT.
SDK-51857	715638	All 56440_A0 56440_A1 56440_B0	An error with _bcm_esw_stat_flex_destroy_ingress_table_counterswas reported. Fixed bug in stat_group_create which was intializing '256' counters.
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-51900		56640_B0	In earlier releases, priority to queue mapping was limited to 15 entries.The table supports 16 entries per profile. There was an error in validating the number of entries parameter which was corrected
SDK-51902	705911	56440_A0 56440_B0	Corrected issue with BFD event thread not exiting by increasing the timeout to 5 seconds.

Table 106:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.0*
SDK-51920		88660_A0	OAM/BFD events: Support the DMA reroute writes intended to the Interrupt Message Register to a local host memory. To support this functionality the following soc properties should be configured: 1) oamp_fifo_dma_enable - enables fifo dma mode. Default is 0. 2) oamp_fifo_dma_buffer_size - length of the messages buffer we store in the CPU. 3) oamp_fifo_dma_timout - the time for generating an interrupt when the fifo is not full. Value 0 indicates interrupt is sent only when fifo is full. Default is 0 4) oamp_fifo_dma_threshold - the number of events written until interrupt is generated.
SDK-51925	702621	88650_A0 88650_B0 88650_B1	Trill allows now multiple flooding-groups with the same nickname. This can be used to create flooding with the same nickname for different VSIs. Procedures which used both nickname and ID as the key (like trill_port_get) will work only with id. See an example of configuration in cint_trill: trill_with_two_vlan_flooding.
SDK-51933		88660_A0	In stacking systems, BCM88660 is able to pass 16 bits of Load-Balancing key and to reproduce the hashing decision in the second system. Thus, any limitation or performance decrease in hashing trunk is removed when using this option. In the data path, the first LB-Key byte is copied in the FTMH.LB-Key-Extension, whereas the second byte is copied in the second User-Header. At egress editor block of the first system, the second User-Header is copied to the start of packet to be extracted by the Ingress PMF in the second system. This option can be enabled by setting the following SOC properties: 1. system_ftmh_load_balancing_ext_mode=FULL_HASH 2. first_header_size_<all stacking ports>=1 3. field_class_id_size_1=8
SDK-51934		88650_A0	In Field Processor, the Direct table is one of possible Databases (bcmFieldGroupModeDirect). Its key is very short (10 bits maximum) and corresponds to the index line of the TCAM Action table. The support of the Warmboot was faulty in the Driver: the entry content was not retrieved correctly. This is fixed.
SDK-51939	717396	All 56850_A0 56850_A1 56850_A2	Modify bcm_port_queued_count_get() to support in Trident2
SDK-51961	712277	88650_B0	MIM: DEFAULT BEHAVIOR CHANGE . bcm_l2_station_get() API failed in some cases when VSI for MyMac was considered to be B-MyMac. This happened when the MIM indication 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 indication bit. The MIM indication bit in station_id changed from bit 7 to bit 29.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-51984	711243	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 earlier releases, an issue was reported in soc_tr_parity_process_mmu_qcn(). Corrected the bit number values for detecting the memory type.
SDK-51993		88660_A0	Trill: Introduce Multi-homing connectivity to the trill network allow a host to have access to trill campus using more than one RBridges. The host treats a group of edge RBridges as an Uplink link bundle that works in an active-active load sharing model. Arad+ support up to 3 virtual RBridges in system. See cint_trill_multi_homing.c for more description and packet-flows
SDK-51994	707370	88650_B1	IP tunnel termination lookup key is defined by SOC property: bcm886xx_ip4_tunnel_termination_mode. Added 2 new lookup key for IPV4 tunnel termination: bcm886xx_ip4_tunnel_termination_mode = 4 - Key is : {DIP, SIP, IPV4.Next-protocol} bcm886xx_ip4_tunnel_termination_mode = 5 - Keys are : {DIP, SIP, IPV4.Next-protocol}, {DIP} Lookup IPV4 next protocol is useful to configure multiple separate VPNs, with same DIP and SIP, but with different tunnel-types. See an example of use in: cint_ip_tunnel_term.c
SDK-52013		56440_A0 56243_B0 56240_B0 56242_B0	1.Corrected configuration of shared pool sizes for Saber. 2.Only ports 25 tot 28 of MXQPorts use PG7. SDK was setting it for ports 25 to 34. This is corrected now. 3.Corrected configuration of RQE_WQE, CFAPI, CFAPE and QENTRY free address pools based on how devide is OTPd. This will prevent ECC errors when using Saber(BCM56240).
SDK-52033		56150_A0	Fixed DXGS mode of HG ports may not be consistently programmed.
SDK-52081		56850_A2	The JIRA fixes the temperature reading bug in the TSC driver. Also it provides the chip version information in DSC dump.
SDK-52098		56850_A0	Renamed BCM_FCOE_VSAN_NORMALIZED_CHECK to BCM_FCOE_VSAN_NORMALIZED_ZONE_CHECK
SDK-52110	720063	88030_A0	Support the encoding and decoding of ITMH, NPH and OAM headers in cint.
SDK-52139	679766	56850_A0 56850_A1 56850_A2	In earlier releases Port Extender Controlling Bridge not working consistently in a stacked configuration. This has been resolved. On the egress chip, we must program the SOURCE_VP table entry with TPID_SOURCE=2, i.e. use TPIDs based on SGLP. The default is 0 - use SVP-based TPIDs from this table. mod source_vp 1 1 TPID_SOURCE=2 Once we do this, the stacking setup for port extender works.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-52148	718595	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	Problem: bcm_field_action_add() API in SDK uses only one part of a FP_POLICY_TABLE entry to program the Field actions applied to an entry. bcmFieldActionCosQCpuNew and bcmFieldActionServicePoolIdNew are conflicting actions for SINGLE WIDE entry. For DOUBLE WIDE entry both actions can be applied to an Entry by programming them independently in the two parts of FP_POLICY_TABLE belonging to a double wide entry. But software does not support this behavior and bcm_field_action_add() API must be enhanced to support the same. Solution: Enhanced bcm_field_action_add() API to program bcmFieldActionCosQCpuNew and bcmFieldActionServicePoolIdNew Field Actions in different parts of an Entry belonging to a Multi Wide Group as hardware support this feature. For an entry belonging To Single Wide mode group, API will return BCM_E_CONFIG error as these two actions are conflicting and must be programmed in the same hardware field in FP_POLICY_TABLE.
SDK-52166	715996	88650_A0 88650_B0 88650_B1	In External TCAM, when used with forwarding tables, each entry content (data and mask) is saved internally in a hash table to get the KBP Driver Entry-ID. This hash table had an incorrect key-size (only according to data), thus 2 entries with same data but different masks were considered as being identical. This is fixed.
SDK-52168		88750_A0	Minor change to FE1600 isolation sequence, no effect on functionality or customer application.
SDK-52169		88650_A0 88650_B0 88650_B1 88660_A0	PON: bcmVlanPortIgnoreInnerPktTag can be used only for PON-Ports. NNI/CPU/Recycle ports do not support that vlan control value.
SDK-52189		88650_A0 88650_B0 88650_B1	The ISSU version handling is fixed. Otherwise, 6.3.4 would not be ISSU-able from 6.3.3.
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-52234	720648	All 56850_A0 56850_A1 56850_A2	In earlier releases L3_IIF_PROFILE table profile sharing was not working correctly. In this release we have.added logic to find a matched entry in l3_iif_profile table during update of a entry.
SDK-52241		56850_A0 56850_A2	This fix modifies driver code to support HG20G non-scramble mode with DFE off. The scramble mode requires the DFE to be on.
SDK-52247	721059	56643_A0 56644_A0 56643_A1 56644_A1	Added fix to update the field group selector (IFP) during warm boot if vpn qualifier is part of the qset.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-52253	716433	56850_A0 56850_A1 56850_A2	[ALPM] Some of the ipv4 streams were not hitting an LPM route. If there is a more specific match (in case of a bucket miss) for a destination IP, in some cases the more specific match could miss and hit the global default. This change fixes that issue.
SDK-52264	721288	56850_A0	Counter XAUI activity feature support for TD2 has been removed. `_soc_xgs3_update_link_activity` will not be called for TD2.
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-52358	722565	56850_A0	Support to get rtag7 hash value in port based HiGig proxy mode has been added.
SDK-52361	722981	56850_A2	Added L3 lock in ser correction call to avoid deadlock in mem op and dpc ser correction thread. Moved definition of L3 lock from BCM to SOC layer.
SDK-52362	723016	88650_A0	PON: Creation of VLAN-Port with port parameter as VOQ-PON resulted in the API failure in the case where the PON PortnProfile that associate to the VLAN-Port is not profile 0. The sequence to support it 1.create VOQ per destination system-port (PON-port 0-7) `2.bcm_vlan_port_create` with port being flow-VOQ gport. API will retrieve the correct PortnProfile and update the learn-information of PON-LIF to be VOQ
SDK-52368	721631	88650_A0	`cint_vlan_control_config.c` CINT example missing documentation specifying that `dflt_frwrd` variable must be set to 1 in ARAD/ARAD+. VSI flooding group must be set the same for all unknown-uc/unknown-mc/broadcast fields. In order to set various default forwarding modes, e.g unknown unicast, unknown multicast and broadcast, use: bcmPortControlFloodUnknownUcastGroup, bcmPortControlFloodUnknownMcastGroup, bcmPortControlFloodBroadcastGroup
SDK-52381	717920	56850_A0	In earlier releases, L3 Conflict Get, `bcm_td2_l3_conflict_get()` was broken. This has been resolved.
SDK-52397	722792	88030_A0	Fixed bcm88030 A1 port status LED issue
SDK-52405	723353	88650_A0 88650_B0 88650_B1	RX-LOS application - added support interlaken ports
SDK-52407	723478	56850_A0	Added support for Concatenate mode in calculating ECMP, LAG and HGT rtag7 hash index.
SDK-52416		88660_A0	ARAD+ fails to init OAM after WB when adding lm or dm object.
SDK-52419		All	Added a check to prevent statistics increment if replace and ID flags are set

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-52434 SDK-47421	723350	56640_A0 56640_A1 56640_B0	Fixed packet alignment issue on higig port. When a higig port is connected to external phy and if user configures 42K speed on it, then xlgmii_align bit should be set with 1. This fix sets the xlgmii_align bit when higig port is configured in 42000 speed.
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-52453		88650_A0 88650_B0 88660_A0	In Field Processor, external TCAM configuration was not restored during warmboot. Preliminary support is added to restore external TCAM configuration during warmboot. Warmboot is not supported for External TCAM yet.
SDK-52469	701853	All	BCM_FIELD_DATA_QUALIFIER_OFFSET_NEGATIVE is not valid on xgs ,hence returning BCM_E_UNAVAIL when qualifier is set
SDK-52471	723924	88030_A0	Order issue addressed in template generated code.
SDK-52474		56850_A1 56850_A2 56850_A0	Added doc changes for bcmFieldActionETagNew [Add/Change ETAG] & bcmFieldActionETagDelete [Delete ETAG] in IFP to support Port Extenders Etag add/delete/change options.
SDK-52490	724657	56640_A0 56440_A0 56440_B0	In earlier releases it was not possible to send a one-step 1588 packet via PCIE. In this release support has been added to specify the timestamp offset during CPU packet tx.
SDK-52496	723483	84756_A0 84756_C0	An issue was reported where hot swap on 8x10G card failed. A fix was required in the phy84756_fcmap.c driver : Added pluggable PHY support
SDK-52512		88650_A0 88660_A0	MPLS VPN creation bcm_mpls_vpn_id_create is now valid for both VPN ranges 0-4K and 4K-32K.
SDK-52513		88650_A0	VLAN-Port: SW-DB forwarding information of VLAN-Port might not update correctly because of uninitialized parameters caused bcm_vswitch_port_add to fail on random cases.
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-52518	705177	88650_B1	STG Warmboot: Warmboot stored STP state per port up to STG-ID 12 and so didn't recover for all other groups. Updated Warmboot STG allocation size to correct size.
SDK-52519		88750_B0	Snake test with external loopback failed when running it over BCM88750_B0. Fixed.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
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-52526	719683	88650_A0 88650_B0 88650_B1 88660_A0	VLAN: A VLAN-Port object can be created per port by calling the API `bcm_vlan_port_create()` with a `MATCH_PORT` criteria and can be identified by a `vlan_port_id` value. The object may be destroyed using `bcm_vlan_port_destroy()` by suppling the `vlan_port_id.` Destroying the object frees the `vlan_port_id` that can be used for some other VLAN-Port object creation when the `WITH_ID` flag is used and the `vlan_port_id` is supplied. A problem occurs when performing a create and destroy sequence for 3 times with the same `vlan_port_id.` The third creation fails as some resources weren't freed correctly during the destroy of objects with `MATCH_PORT` criteria. The resource freeing during destroy, was fixed for the `MATCH_PORT` criteria objects as well. The issue detailed above affects the unicast RPF mode per RIF feature (the SOC property `bcm886xx_l3_ingress_urpf_enable=1)` `.` When this feature is used by specifying a uRPF mode other than `BCM_SWITCH_URPF_DISABLE` in `bcm_l3_ingress_t.urpf_mode` for `bcm_l3_ingress_create,` deleting LIFs which are members of RIFs that use uRPF with the `MATCH_PORT` criteria will result in undefined behavior.
SDK-52529		88660_A0	Support oam accelerated loopback. See an example of use in : `cint_oam_arad_plus.c`
SDK-52575	725460	53343_A0 56150_A0	Correct supported number of multicast replication interface of BCM56150.
SDK-52583	710089	56450_A0	CLI command "oam endpoint show" fixed to exhibit correct endpoint information for BCM56450
SDK-52584	725729	56450_A0	In the previous release SP and WRR did not work correctly on extended queues. The reason was that the MMU thresholds were not configured for extended queues. Added queue configuration assuming internal-lossless settings
SDK-52588	725824	56450_A0	Fixed to handle -1 as numq for scheduler gport.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-52592		88650_A0 88650_B0 88650_B1 88660_A0	MPLS VPNs can be created using bcm_petra_mpls_vpn_id_create(). In DNX Arch only VPLS mode requires valid VPN ID. VPWS does not require to go by VPN (Cross-connect only). The VPWS VPN creation isn't supported but the API succeed to allocate a VPN that in fact is created with VPLS encoding. For the 6.3 branch, the ability to enter VPWS VPN is valid only for ID 0 For 6.4, the VPWS flag is no longer supported, as it is not required in MPLS VPWS sequence. See cint_vswitch_cross_connect_p2p.c for more information on VPWS connection.
SDK-52600		88650_A0 88650_B0 88650_B1 88660_A0	A LIF can be created with no Ingress AC-Key matching by calling bcm_vlan_port_create() with a criteria field set to BCM_VLAN_PORT_MATCH_NONE. Later, the LIF values may be edited by calling the same BCM API with the additional BCM_VLAN_PORT_REPLACE flag. The modification of LIFs (using BCM_VLAN_PORT_REPLACE) that were originally created with criteria BCM_VLAN_PORT_MATCH_NONE, was failed.
SDK-52618	724270	88030_A0	SDk-52618 RCE Errors (ECC etc) - was test packet
SDK-52622		56850_A1	An issue was reported with the performance of the L2 match replace function. 1. Resolved RPC issue for the new API array arguments by fixing the papi. 2. Updated internal API implementation using bitmaps instead of multiple iterations to improve the performance.
SDK-52628		88660_A0	CGE1 traffic fall in NBI in case of 2Caui +Elk was fixed
SDK-52629		88660_A0	Fixed: PRBS APIs support 2 CAUI + ELK
SDK-52630	726283	56450_A0	Changed the delete sequence for L0 and L1 Nodes First delete the subtree node first and then delete the parent node
SDK-52633		88650_A0 88650_B0 88650_B1	In 6.3.3, a version compiled without the WARMBOOT compilation flags was failing at initialization due to minor code missing in switch init. This is fixed.
SDK-52637		88650_A0 88660_A0	In Policer, a single 2-rate color blind meter with the default configuration (32 range mode, SERIAL) would not do rate limitation. This behavior is now corrected.
SDK-52639	710412	88650_A0 88650_B0 88660_A0	In Egress L2, an HW field (CustomLearn) was set by mistake - it is a debug-only not-validated field intended to allow more packets to be learnt.
SDK-52643	723104	88650_B1	In 1588 application, an ITMH packet can be injected with an OAM-TS header above to indicate the header offset for the timestamp. Due to HW implementation, the usage of User-Header requires the following action from the user: - if a SOC property field_class_id_size_X is set, then an injected packet of type Ethernet over OAM-TS over ITHM over PTCH-2 requires: 1. to insert the User-Header to be inserted between Ethernet and OAM-TS 2. the Opaque-Attribute.LSB in PTCH-2 (bit 12) must be set 3. the ITMH destination is of type System-Port

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-52667	724073	88750_A0 88650_A0 88650_B0 88650_B1	Fixed a misconfiguration when setting FE1600 to work in repeater mode, that could cause occasional drops.
SDK-52668	725913	56850_A0 56850_A1 56850_A2	Fixed issue where hash-move when moving an invalid entry may break wider conflict entry.
SDK-52673	726396	56850_A0 56850_A1 56850_A2	an issue was reported where Buffers were getting stuck on control queue of egress port when port was shut in the presence of fragmentation. This has been resolved as described below: When port is disabled or link down, remove related bitmap from below three registers. Thus traffic will not be enqueued to these ports any more. THDU_OUTPUT_PORT_RX_ENABLE0_64 MMU_THDM_DB_PORTSP_RX_ENABLE0_64 MMU_THDM_MCQE_PORTSP_RX_ENABLE0_64
SDK-52678		88750_A0 88650_A0 88650_B0 88650_B1	In a multi-stage system, live removal of an FE1600 or ARAD device could result in occasional drops. Fixed.
SDK-52691	726146	88650_A0 88650_B0 88650_B1 88660_A0	VLAN-Port Advanced VLAN translation: Packet discard can be set either per physical port or per LIF. In standard VLAN edit mode, the API bcm_port_discard_set() is used to set the discard state both for physical ports and for Out-LIFs. In Advanced VLAN edit mode, the API bcm_port_tpid_class_set() is used to set the physical discard state, while Out-LIF discard is also blocked in bcm_port_discard_set(). The API bcm_port_discard_set() is now available for setting Out-LIF discard state in AVT mode as well. The same way, bcm_port_discard_get() now retrieves an Out-LIF discard state in AVT mode.
SDK-52699	725215	88650_B1	Fabric source routed cell receive did not support multiple SR cells in parallel. Fixed.
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-52731		88650_A0 88650_B0 88650_B1	Fixed corrective action in case of parity error interrupt in WDF table
SDK-52733		56850_A0 56855_A0 56854_B0 56854_A0	In earlier releases ROUTE updates were decreasing bcmSwitchObjectEcmpCurrent count in error. This has been addressed.
SDK-52741	720579	88650_A0 88650_B0 88650_B1 88660_A0	Fast flush enables clearing MACT entries for LIFs that are associated with a ring protection group FEC using bcm_l2_replace() using the BCM_L2_REPLACE_PROTECTION_RING flag. A LIF association to a group is done by calling bcm_port_class_set() with the class set to bcmPortClassL2Lookup. A LIF association of a remote LIF to ring protection group performs LIF HW configuration instead of only SW DB configuration, resulting potential problems when remote LIFs are used. The API bcm_port_class_set() has changed so that it configures the LIF HW only for local LIFs. Ring Protection CINTs and tests where updated to support a multidevice setup. See cint_l2_fast_flush.c

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-52742		88650_A0 88660_A0	Support TDM (OTN/CBR) traffic for mixed systems. Specially for mixed systems with ARAD and PetraB, when using VCS256 fabric cells, the packets are split to smaller cells by fabric devices (FE1600) during their transport, and are reassembled at the receiving FAP. The reassemble is done based on an attribute called "source-FAP-ID". This source FAP-ID is calculated as the FAP-ID of the source FAP plus a configurable offset. This source-FAP-ID value must be unique in the TM domain, and different from all FAP-IDs in the TM domain. The offset is specified by a new soc property called `tdm_source_fap_id_offset` . If it is not specified by the soc property, its default value is 256.
SDK-52758	727046	88030_A0	BCM88030: fixed bug where MPLS label was over-writing the IP address RCE key field.
SDK-52762	728229	56450_A0	CoE/LinkPHY subports are not added to default VLAN 1. Customer application needs to manage the VLAN 1 membership for CoE/LinkPHY subports. The support should be added to the VLAN 1 after creating the scheduler tree for the subport.
SDK-52763	719360	88650_A0 88650_B0 88650_B1 88660_A0	Fixed mirroring and snooping settings that did not work in certain cases.
SDK-52766		88650_A0 88650_B0 88650_B1 88660_A0	When a CAUI port is disabled and then enabled (no traffic is running), the MIB counters show incorrect values. The issue was fixed.
SDK-52767		88750_A0 88650_A0 88750_B0 88650_B0 88650_B1 88660_A0	"show features" diagnostics was added.
SDK-52771		88650_A0 88650_B0 88650_B1	RSPAN: does not work when port control bcmPortControlErspanEnable is set to 1
SDK-52772		88650_A0 88650_B0 88660_A0	OAM snooped packets are corrupted - snooped packet arrives to the CPU with trap headers.
SDK-52774	728360	56850_A0	In earlier releases, deleting and adding back source to mtp dest in TX direction returned -14. This has been addressed by clearing the egress mirroring MTP index of port unless the MTP slot is released.
SDK-52782	728467	56850_A0	In this release, we can use `DISABLE_TUNNEL_IP4_GRE_IP6` `DISABLE_TUNNEL_IP4_GRE_IP4` to set `DISABLE_HASH_INNER_IPV4_OVER_GRE_IPV6_A/B` `,DISABLE_HASH_INNER_IPV4_OVER_GRE_IPV4_A/B` individually, or still use the old flag `BCM_HASH_FIELD0_DISABLE_TUNNEL_IP4_GRE` to set both fields in `RTAG7_HASH_CONTROLr` as legacy. IPv6 flags are same to IPv4.
SDK-52788	728597	All	Solved `FIELD_ENTRY_MISMATCH` problem in `bcm_field_qualify_IpType_get` by implementing new device specific functions to get iptype encoding using `hw_data` and `hw_mask`.
SDK-52795	728851	56447_B0	MMU threshold settings for extended queues has been updated for Katana (BCM5644x)
SDK-52796	728261	56450_A0	In previous releases ICAP packet resolution L3MCKnown did not work. In this release we have fixed IFP packet resolution for BCM56450

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-52806	728092	56648_A0	On change of priority for the VLAN, A small delay was introduced as new priority mapping happens after deletion of old index. On live traffic, this transitional delay caused some packets to be on priority 0 queue as the mapping is under transition. Fixed in the transitional delay for smooth traffic flow on priority transition.
SDK-52821		56850_A0	Updated documentation for the following new API;s added. bcm_cosq_stat_sync_get, bcm_cosq_stat_sync_get32
SDK-52823		56850_A0	New API's added for cosq_stat retrieval have been added. bcm_cosq_stat_sync_get bcm_cosq_stat_sync_get32. Similar to bcm_cosq_stat_get(), value returned is software accumulated counter synced with the hardware counter.
SDK-52830		88030_A0	Fixed taps unified mode ucode lookup issue with certain config on bcm88030
SDK-52831		88030_A0	Fixed taps capacity resource leak update rate and host memory leak issue found in last patch on bcm88030
SDK-52833 SDK-52190		56850_A2	This JIRA fixes the problem that TSCMOD doesn't have a clean restart for CL72, such that it may cause incorrect training results. The probablity of the training problem is about 1% of link restart.
SDK-52836	728502	All	Fixed Assert While installing VFP entries with flex counters attached
SDK-52844	728330	56450_A0	Code fixed to update the reference count properly (next hop entry used by mpls port)
SDK-52857	695985	All	Updated grog for Inports to reflect the correct behavior
SDK-52859	707972	54680_A0 54682E_A1	Enhancing documentation/description for the config property phy_port_primary_and_offset_<port>
SDK-52860	728139	88650_B1	In some scenarios, dependent on the allocated ports and typically involving CAUI ports, an underrun may occur resulting in not reaching full port capacity. Fixed.
SDK-52873	729725	56450_A0	Fixed the following issues for BCM56450 1. programming of FP_DOUBLE_WIDE_SELECT.slice_x_f1. 2. proper initialization of IFP_SINGLE_WIDE_F1_5.
SDK-52895	729741	All	RPC has been enabled for the HASH bank APIs.
SDK-52918	729962	56840_A0	Added support for handling remote fault link status.
SDK-52923		88660_A0	OAM: New support in updating loss and delay objects was added. New object is created using bcm_oam_loss/delay_add(). The update is performed using the same api with the flag BCM_OAM_LOSS/DELAY_UPDATE flag set.

Table 106:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.0***
SDK-52924		`88650_A0 88650_B0 88650_B1 88660_A0`	MPLS: A VSI can be associated as an MPLS VPN by calling `bcm_mpls_vpn_id_create()`. The VPN ID should be supplied as well as a `BCM_MPLS_VPN_WITH_ID` flag. Changing VPN fields is possible after creation, using the additional `BCM_MPLS_VPN_REPLACE` flag. The same VSI may also be used for vswitch, MIM, etc. The handling of the `BCM_MPLS_VPN_WITH_ID` & `BCM_MPLS_VPN_REPLACE` flags wasn't correct. Performing a create, with the replace flag `BCM_MPLS_VPN_REPLACE`, failed for an allocated VPN ID instead of succeeding. The same way, for an unallocated VPN ID, the API succeeded instead of failing. The behavior of the `BCM_MPLS_VPN_WITH_ID` & `BCM_MPLS_VPN_REPLACE` flags in `bcm_mpls_vpn_id_create()` was fixed.
SDK-52925		`88660_A0`	arad+ : minimum number of links warning message will be displayed only when current number of links < minimum number of links configured
SDK-52930	728932	`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 it was found that if you set bcmPortControlMmuDrain control to an admin-down port (disabled port), then after the port is re-enabling, the traffic to that port will be blocked. This issue was fixed by restoring the value of `XLMAC_CTRL` after draining cell.
SDK-52943	730095	`56840_A0`	New option "nocache" added to dump command, to display h/w table contents skipping cache.
SDK-52951		`88650_A0`	Enabling of slow rate 2 corrected: Previously, calling: `bcm_cosq_control_set` with control: bcmCosqControlFlowSlowRate and arg=1 would have set slowRate2. If the same API was called with the same control and arg=2 an error would occur. Both were corrected so now calling with arg=0 disables `slow_rate`, calling with arg=1 enables `slow_rate1` and calling with arg=2 enables `slow_rate2`
SDK-52952		`88650_A0 88650_B0 88660_A0`	OAM-BFD co-existence: `bcm_bfd_init` resets some of the oam registers (in `bcm_oam_init)`. Thus no oam endpoints can be added before calling `bfd_init`.
SDK-52954	723963	`88650_A0`	For better latency performance, especially when working with high volume traffic, ilkn ports can be configured to have dedicated TDM queues at the egress. This feature is enabled by setting the soc property: `ilkn_tdm_dedicated_queuing=1`.
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.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-52971		88660_A0	ARAD plus device supports either BFDoPWE or BFDCCoMPLSTP encapsulations. This should be defined by the user with a soc property. `bfd_encapsulation_mode` soc property is setting bfd pwe (mode 0) or bfd cc mplstp mode (mode 1). 0 by default. See example of use in `cint_bfd.c.`
SDK-52990	730016	88650_A0 88650_B0 88650_B1 88660_A0	Advanced VLAN translation: Upon Configuration of an advanced VLAN edit action entry using `bcm_vlan_translate_action_id_set()`, the TPID values that will be used are mandatory fields. At the Ingress side, those TPIDs are matched with an existing TPID profile value that is passed to the Egress. The API is failed if no such TPID profile matching is found. The TPID profile consists of up to two global TPIDs. The TPID profile matching consists of two stages: 1. Exact match - The supplied TPIDs are similar to those of the TPID profile and are similarly positioned (Outer/Inner). 2. Opposite match - The supplied TPIDs are similar to those of the TPID profile but are inversely positioned (Outer/Inner). The problem occurs when supplying two similar TPIDs: Only a TPID profile that consists of two instances of this TPID will be matched. If no such TPID profile exists, the command will be failed. A third matching lookup was added to address the cases where two similar TPIDs are supplied. This lookup requires that this TPID will be included only once in a TPID profile, in any position, in order to match the TPID profile.
SDK-53007		88650_A0 88650_B0 88660_A0	`bcm_oam_opcode_map_set/get` is now functional
SDK-53011		88650_A0 88650_B0 88650_B1 88660_A0	`bcm_port_learn_set` now supports enable/disable SA when destination is Flow-ID
SDK-53012	715940	88030_A0	Fixed the wrong action type for mirror & drop in egress RCE action table.
SDK-53019		88660_A0	OAM: Add support for `bcm_oam_loss_get`, `bcm_oam_delay_get`, `bcm_oam_loopback_get` apis.
SDK-53021	720668	56850_A0	Updated the documentation related to `BCM_PORT_CONGESTION_CONFIG_DESTMOD_FLOW_CONTROL`
SDK-53045	730837	56540_B0	Added new redirect soc APIs for autoneg and loopback configurations to get accessed for phy specific GPORTs. Implemented this into existing APIs `bcm_port_autoneg_set/get()` `bcm_port_loopback_set/get()` APIs.
SDK-53056		88650_A0 88650_B0 88650_B1	PON: Recycle and mirror ports must be allocated from port 128 and above. Other ports (0-127) are used for PON side.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
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.
SDK-53082	730548	All	Prevent potential data corruption after KNET kernel driver call to skb_padto.
SDK-53099		88650_A0 88650_B0 88660_A0	Trill RPF-Check: In SDK 6.3.3 RPF check was moved to LEM + PMF. See cint_trill.c for more information. In SDK 6.3.4, 6.4.0 removed unused code bcm_trill_multicast_source_add/ bcm_trill_multicast_source_delete/ bcm_trill_multicast_source_get for ARAD.
SDK-53108		88650_A0	Different ports can now be set with pfc/llfc (could not be set differently before). Also - disabling one port fc will not stop fc in a device level - Fixed
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.

Table 106:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.0***
SDK-53129	731105	56850_A0 56850_A1 56850_A2	Fixed link interrupt miss issue in 1G mode
SDK-53149		88650_A0 88650_B0 88660_A0	Fixed wrong check pon port when setting and getting bcmVlanPortIgnoreInnerPktTag for pon port.
SDK-53183	728584	All	API bcm_cosq_gport_connection_get returned error for ISQ ports. The API is now valid also for ISQ ports.
SDK-53192		88650_A0 88650_B0 88650_B1	L3 APIs replace: 1.Support to replace vrf, mac addr, ttl, mtu and dscp qos map id by bcm_l3_intf_create with BCM_L3_REPLACE flag. 2.Support to replace l3 intf, next hop mac, port tgid and encap id by bcm_l3_host_add with BCM_L3_REPLACE flag. 3.Support to replace l3 intf and port tgid by bcm_l3_route_add with BCM_L3_REPLACE flag.
SDK-53194		88650_A0 88650_B0	Support the replace for bcm_mim_vpn_create and bcm_trill_vpn_create.
SDK-53195 SDK-53298		88650_A0 88650_B0 88650_B1 88660_A0	1.Support to replace mc-group and L3 route interface by bcm_ipmc_add with BCM_IPMC_REPLACE flags. 2.Support to replace unknown unicast mc-group, unknown multicast mc-group and broadcast mc-group by bcm_l2gre_vpn_create with BCM_L2GRE_VPN_REPLACE,BCM_L2GRE_VPN_UNKNOWN_UCAST_REPLACE, BCM_L2GRE_VPN_UNKNOWN_MCAST_REPLACE, BCM_L2GRE_VPN_BCAST_REPLACE flags 3.Support to replace match port and flag with/without BCM_L2GRE_PORT_NETWORK by bcm_l2gre_port_add with BCM_L2GRE_PORT_REPLACE flag. 4.Support to replace unknown unicast mc-group, unknown multicast mc-group and broadcast mc-group by bcm_vxlan_vpn_create with BCM_VXLAN_VPN_REPLACE,BCM_VXLAN_VPN_UNKNOWN_UCAST_REPLACE, BCM_VXLAN_VPN_UNKNOWN_MCAST_REPLACE, BCM_VXLAN_VPN_BCAST_REPLACE flags. 5.support to replace match port and flag with/without BCM_VXLAN_PORT_NETWORK by bcm_vxlan_port_add with BCM_VXLAN_PORT_REPLACE flag.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-53196		`88650_A0 88650_B0 88650_B1`	MPLS APIs Replace: 1.Support `bcm_mpls_port_add` to replace `match_label,` vpn, and flags `(BCM_MPLS_PORT_CONTROL_WORD, BCM_MPLS_PORT_ENTROPY_ENABLE)` with `BCM_MPLS_PORT_REPLACE` and `mpls_port_id.` In case of `mpls_port_id` is protected, `egress_label.label` and port of FEC can also be replaced. 2.Support `bcm_mpls_tunnel_initiator_create` to replace label, vsi and action with `BCM_MPLS_EGRESS_LABEL_REPLACE` and `tunnel_id.` 3.Support `bcm_mpls_tunnel_switch_create` to replace `egress_label, qos_map_id, tunnel_if` and flags `(BCM_MPLS_SWITCH_FRR, BCM_MPLS_SWITCH_LOOKUP_SECOND_LABEL, BCM_MPLS_SWITCH_ENTROPY_ENABLE, BCM_MPLS_SWITCH_TRAP_TTL_0, BCM_MPLS_SWITCH_TRAP_TTL_1, BCM_MPLS_SWITCH_SKIP_ETHERNET, BCM_MPLS_SWITCH_NEXT_HEADER_L2, BCM_MPLS_SWITCH_NEXT_HEADER_IPV4, BCM_MPLS_SWITCH_NEXT_HEADER_IPV6)` with `BCM_MPLS_SWITCH_REPLACE` and `in_label,` in case of action is `BCM_MPLS_SWITCH_ACTION_POP.` 4.Support `bcm_mpls_vpn_id_create` to replace `broadcast_group, unknown_multicast_group` and `unknown_unicast_group` with flags `BCM_MPLS_VPN_VPLS|BCM_MPLS_VPN_WITH_ID.` The replaced `broadcast_group` should be equal to `unknown_multicast_group` and also equal to `unknown_unicast_group.`
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().`
SDK-53201	710888	`88650_A0 88650_B0`	ARP extender provides the ability for IPV4 UC packets MAC extension offset from IPV4 host table to next-hop mac address. In ARAD-A/B ARP extender is implemented using the egress-editor micro-code. Program caused on some Trill packets to drop. Modified ARP extender program to handle only IPV4 UC packets as it should be.
SDK-53202	727655	`88650_B0`	`bcm_bfd_endpoint_create WITH_ID` ignores the given id and returns a new allocated one.

Table 106:

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-53225		88650_A0 88660_A0	VLAN: SDK/src/examples/dpp/cint_vlan_translation_new_mode.c was renamed to appropriate name: SDK/src/examples/dpp/cint_advanced_vlan_translation_mode.c
SDK-53227	733542	56450_A0	Corrected code for Multicast traffic. PID will be updated by cosq scheduler function at run time (for subport queue configuration etc)
SDK-53242		88650_A0 88660_A0	VLAN: The api bcm_petra_vlan_control_port_set is responsible of setting miscellaneous port-specific vlan options. It receives as a parameter bcm_vlan_control_port_t type. Two enumerations of this variable are not supported in AVT mode: bcmVlanPortPriTaggedDrop and bcmVlanPortTranslateKeyFirst. Hitherto this change, these cases returned BCM_E_NOT_FOUND. The fix returns BCM_E_UNAVAIL in case the type equals one of the two.
SDK-53255	728560	56640_B0	In the previous release,when external phy called the speed notify of 100M on the internal serdes, the default was 100FX(fiber mode). However 100FX was not supported on warpcore C0, so packets would not go through. Since WarpCore C0 does not support 100FX and the default mode is fiber, so for 100M speed it will not work. The fix will be using sgmii 100M mode for warpCore c0.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-53283	733471	56450_A0	Clear HQOS configuration while switching from extended queuing to diffserv queuing.
SDK-53286	733518	88650_A0 88660_A0	PON: Add a new criteria "BCM_VLAN_PORT_MATCH_PORT_TUNNEL_PCP" to classify PON InLIF based on PON-Port, Tunnel-ID and outer PCP.
SDK-53289	734160	56450_A0	Fixed SOURCE_TRUNK_MAP_MODBASE and SOURCE_TRUNK_MAP configuration for BCM56450. Previously there was a configuraton error for subport.
SDK-53290		88660_A0	When a link status changed , fabric min number of links feature might not work, and traffic won't stop. Fixed.
SDK-53301 SDK-53047		88650_A0 88660_A0	Trill functionality always enabled Trill designated VLAN check: A single VLAN allowed for Trill encapsulated packet on a specific port. Device supports up to 8 different designated VLANs. In case more than 8 different designated VLANs are needed, user needs to disable this check. New soc property added to disable this check - trill_designated_vlan_check_disable=1. User can mimic check using ACLs.
SDK-53304		88660_A0	Introduce a new VLAN-Port property : FORWARD_GROUP In the regular VLAN-Port settings MACT forward to VLAN-Port. In order to forward to VLAN-Port MACT needs two fields information: Out-LIF (outgoing logical interface) and Out-Port (physical destination). Forward-Group allows instead of using the MACT to result for forwarding information, have indirection group (forward-group) to provide the information on destination and the other bits to use the FP settings in a flexible way. The indirection object (Forward-group) that consist both the physical-destination and the outgoing logical-interface (out-LIF) is implemented in DNX using FEC. To allow such a scheme a flag indicate it BCM_VLAN_PORT_FORWARD_GROUP /* Use forwarding group */ Note: When supporting Forward-Group device must disable HW learning and use only CPU learning.
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-53327		88650_A0 88650_B0 88650_B1	PON: In application level, upstream and downstream should use different MC-ID upon VPN PON service creation.
SDK-53338		56850_A1 56850_A2 56850_A0	Fixed range check for VXLAN VN_ID and L2GRE VPNID during vpn create API

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-53341		`88650_A0`	Calculation of channelized interface shaper is now corrected when calling: `bcm_cosq_gport_bandwidth_set(unit,parent_port,cosq,0,kbits_per_sec,0);` Where `parent_port` is: `BCM_COSQ_GPORT_E2E_PORT_SET(e2e_port,port);` `bcm_fabric_port_get(unit,e2e_port,0,parent_port);`
SDK-53345	734829	`88650_A0 88650_B0 88650_B1 88660_A0`	Warmboot: Performing WB or synchronizing the warmboot DB using `bcm_switch_control_set()` may have caused DB segmentation fault due to a short buffer that handles protection FECs. The buffer size was fixed. In 6.3.3, if protection FECs with values of 8K and above were used or if a warmboot was performed, a device reboot in 6.3.4 is required in order to eliminate any possible memory override issues.
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.
SDK-53356		All	Ensure that KNET DMA abort works correctly on idle DMA channels on CMICe-based devices such as BCM5684x.
SDK-53358		`88660_A0`	OAM LM packets are always counted in ARAD due to HW bug. In ARAD+ LM packets are counted only upon user request (can be configured using `bcm_oam_endpoint_action_set` api).
SDK-53360		All	Fixed potential Tx DMA lockup in KNET kernel module.
SDK-53362		`88650_A0`	In Field Processor, when using direct extraction tables, a segmentation fault may have occurred in some cases when setting qualifiers.This is fixed.
SDK-53363	732741	`88650_A0`	On some operating systems in previous releases, init might fail with segmentation fault in egress editor init. This has been fixed

Table 106:

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-53374		88650_A0 88650_B0 88650_B1 88660_A0	The default range of the credit watchdog was fixed to include all queues. The default range before the fix was one queue - queue zero. A side affect of the driver coming with the previous default and not changing it later is that if the system comes up under traffic, queues may get stuck and require ingress soft-reset.
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.

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
SDK-53431		88650_A0 88660_A0	TM only mode: When bcm.user loads in TM mode, it shouldn't matter what PP soc properties are active, since PP is disabled. There was a problem where setting the Advanced Vlan Translation mode soc property caused a conflict that made bmc.user crash at startup. The problem was fixed, and now the device can start normally with both modes on.
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.
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-53470	736427	56450_A0	Fixed issue with mpls port delete for CoE subport on BCM56450
SDK-53472		88650_A0 88650_B0 88650_B1	In the example application (called by arad.soc), the RX module was not activated after warmboot. This is fixed by calling DPP application in WB mode. Reference code for customer application. No driver change.
SDK-53473	736455	56450_A0	In an earlier release `bcm_cosq_gport_attach` return `BCM_E_RESOURCE` after several rounds of subport add and delete actions. In this release hardware resources of strict priority children are now bing released when the number of children becomes zero which makes node unresolved function consistent with node resolve.
SDK-53478		88650_A0 88650_B0 88650_B1 88660_A0	MPLS: `bcm_mpls_vpn_id_get()` retrieves VPN information from a VSI that was associated as an MPLS VPN `(bcm_mpls_vpn_id_create).` The same VSI may also be used for vswitch, MIM, etc. The supplied VSI to `bcm_mpls_vpn_id_get()` was validated for VSI existance, but it also successfully retrieved VSI info for VSIs that were allocated by other applications, but were not used by the MPLS. The validation for `bcm_mpls_vpn_id_get()` was fixed so that VSIs that are not used by the MPLS, return an `E_NOT_FOUND` error.
SDK-53484		88650_A0	Fixed uninitialized value in internal function `_bcm_dpp_am_egress_encap_init.`

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
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-53531	727653	88650_A0	BFD packets may now be trapped to custom gports. When calling bcm_bfd_endpoint_create(), set the remote_gport field to a valid gport for trapping BFD frames to that gport. Macros such as BCM_GPORT_LOCAL_SET() should be used for converting ports to gports and setting remote_gport. If the default behavior is preferred, remote_gport should be set to BCM_GPORT_INVALID (this is configured in bcm_bfd_endpoint_info_t_init()).
SDK-53542		88650_A0	ECMP - Trunk: a new CINT has been inserted, emulating the 88650 HW and how an hash member is selected in ECMP and Trunk (i.e., LAG). The CINT is called: cint_trunk_ecmp_lb_key_and_member_retrieve.c This CINT does not apply on BCM88660.
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_stat_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.
SDK-53619		88650_A0 88650_B0 88660_A0	Within advanced VLAN mode, cos_profile should be explicitly attached to LIF using bcm_qos_port_map_set() .

Table 106:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.0*
SDK-53633		88650_A0 88660_A0	Warm Boot: Creating an In-LIF object may also create a FEC object as part of a 1:1 protection scheme or a forwarding group of a VLAN Port, or as part of PWE protection. This is done by calling calling `bcm_vlan_port_create()` or `bcm_mpls_port_add()` respectively. If such an operation that creates a protection FEC, was created after a warmboot has been performed, it would fail. The fix, enables the creation of a the above protetcion FECs after performing warm boot, as expected.
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.

Table 106:

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

Table 106:

Number	CSP #	Chips	Release Notes For 6.4.0
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.
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.