Release Notes
For
Switch Software Development Kit

SDK 6.5.9

© 2017 Broadcom. All rights reserved.

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

Table of Contents

Section 1: About This Document

Section 2: Product Documentation

Section 3: New devices added to this release
- Section 3.1: Supported XGS Switch Devices
- Section 3.2: Preview XGS Switch Devices
- Section 3.3: Supported External PHYs
- Section 3.4: Preview External PHYs

Section 4: New Features per Device
- Section 4.1: BCM56970 (Tomahawk2) Updates
  - Section 4.1.1: Device revision support
  - Section 4.1.2: New features supported on BCM56970
  - Section 4.1.3: Known limitations
- Section 4.2: BCM88060 (Montreal2) B0 Beta support
  - Section 4.2.1: Status of supported features
  - Section 4.2.2: Notes and Known Limitations
- Section 4.3: BCM56170 (Hurricane3-MG) A0 Beta Support
  - Section 4.3.1: Features supported
  - Section 4.3.2: New and enhanced features supported on BCM56170 A0
- Section 4.4: Updated External PHY Support
  - Section 4.4.1: BCM84888 (Orca) support
  - Section 4.4.2: BCM82332 (Dino) support
  - Section 4.4.3: BCM54190 (Seahawks) support
  - Section 4.4.4: Interoperability testing for new PHY/switch combinations
    - Section 4.4.4.1: BCM56860 and PHY BCM84328
    - Section 4.4.4.2: BCM56760 and PHY BCM84328
    - Section 4.4.4.3: BCM56760 and PHY BCM84868
    - Section 4.4.4.4: BCM88670 and PHY BCM82322

Section 5: Things to note
- Section 5.1: PSIRT Reporting Process
- Section 5.2: Policy on API defaults
- Section 5.3: Virtual Machine support
- Section 5.4: SDK releases out of active engineering support
- Section 5.5: Warmboot Notes and Considerations
  - Section 5.5.1: Validated Warmboot upgrades
  - Section 5.5.2: Validated Warmboot downgrades
  - Section 5.5.3: Warmboot upgrade/downgrade considerations

Section 6: Summary of BCM API changes and enhancements
Section 6.1: Changes in Class of Service Queue Configuration
Section 6.2: Changes in Explicit Congestion Notification (ECN)
API Changes in Explicit Congestion Notification (ECN)
bcm_ecn_map_mode_t_init
bcm_ecn_map_mode_set
bcm_ecn_map_mode_get
Section 6.3: Changes in Fabric
API Changes in Fabric
bcm_fabric_link_profile_set
bcm_fabric_link_profile_get
bcm_fabric_multicast_local_route_set
bcm_fabric_multicast_local_route_get
bcm_fabric_link_topology_config_set
bcm_fabric_link_topology_config_get
bcm_fabric_rci_resolution_set
bcm_fabric_rci_resolution_get
Section 6.4: Changes in Field Processor
Section 6.5: Changes in Mirroring
Section 6.6: Changes in Operations, Administration, And Maintenance
API Changes in Operations, Administration, And Maintenance
bcm_oam_profile_create
bcm_oam_profile_delete
bcm_oam_lif_profile_get
bcm_oam_lif_profile_set
bcm_oam_profile_action_get
bcm_oam_profile_action_set
API Changes in Policer
bcm_policer_expansion_groups_set
bcm_policer_expansion_groups_get
bcm_policer_database_create
bcm_policer_database_destroy
bcm_policer_control_set
bcm_policer_control_get
bcm_policer_engine_database_attach
bcm_policer_engine_database_get
bcm_policer_engine_database_detach
bcm_policer_database_enable_set

bcm_policer_database_enable_get
Section 6.8: Changes in Port Configuration
API Changes in Port Configuration
bcm_port_priority_config_set
bcm_port_priority_config_get
bcm_port_lane_to_serdes_map_set
bcm_port_lane_to_serdes_map_get
Section 6.9: Changes in Precision Time Protocol
API Changes in Precision Time Protocol
bcm_tdpll_input_clock_ql_change_callback_register
bcm_tdpll_input_clock_ql_change_callback_unregister
Section 6.10: Changes in Packet Transmit and Receive
Section 6.11: Changes in Statistics
API Changes in Statistics
bcm_stat_engine_t_init
bcm_stat_counter_explicit_input_data_t_init
bcm_stat_eviction_t_init
bcm_stat_counter_set_map_t_init
bcm_stat_expansion_data_mapping_t_init
bcm_stat_counter_interface_key_t_init
bcm_stat_expansion_select_t_init
bcm_stat_counter_interface_t_init
bcm_stat_engine_enable_t_init
bcm_stat_counter_output_data_t_init
Section 6.12: Changes in Trunking (Link Aggregation)
API Changes in Trunking (Link Aggregation)
bcm_trunk_find
bcm_trunk_id_set
bcm_trunk_local_attributes_set
bcm_trunk_local_attributes_get
bcm_trunk_local_attributes_t_init
Section 6.13: Changes in BCM types
Section 6.14: Changes in VLAN Management
Section 6.15: Changes in VXLAN Management

Section 7: Test Statistics
Section 7.1: How to read the data
Section 7.2: Overview
Section 7.2.1: Linux kernel versions used in this release

Section 7.3: Total Tests
Section 7.4: API Test Results
Section 7.5: PHY Test Results
Section 7.5.1: SQA External PHY
Section 7.5.2: Interop External PHY
Section 7.5.3: Interop Internal PHY
Section 7.6: Static Code Analysis
Section 7.6.1: Unresolved Static Code Analysis Issues

Section 8: Service Impacting Defects

Section 9: Resolved Issues for 6.5.9

Section 10: Unresolved Issues for 6.5.9

Section 11: Device and Platform Support

Section 12: Compatibility
Section 12.1: Broadcom Task Engines (BTE) Firmware Compatibility Matrix
Section 12.2: BMACSEC SDK Compatibility Matrix
Section 12.3: iMACSEC SDK Compatibility Matrix
Section 12.4: PHY Firmware Compatibility Matrix

Section 13: SDK Externally Licensed Software Components
Section 13.1: EDITLINE License terms and conditions
Section 13.2: LIBXML2 - XML C parser terms and conditions
Section 13.3: ED Editor License terms and conditions
Section 13.4: CINT parser license terms and conditions
Section 13.5: BIGDIGITS license terms and conditions
Section 13.6: APIMODE parser license terms and conditions
Section 13.7: Wind River Systems license terms and conditions
Section 13.8: SFlow license terms and conditions

Section 14: Feedback

Section 1: About This Document

These are the Release Notes for the Broadcom Network Switching Software Development Kit Release 6.5.9.

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

Section 2: Product Documentation

The following documents are available through Broadcom's Customer Support Portal at https://csp.broadcom.com/group/customers/csp. They are the primary source of information and should be referenced when using this release:

Document	***Description***
56XX-PG659-R	*BCM API Reference Guide* This manual describes the theory of operations of the API and all existing BCM APIs for this release.
56XX-PG821-R	*Network Switching Software Platform Guide* This guide describes the SDK source and Makefile structure, abstraction and porting layers, device specific interactions, and the platform/operating system specific features of the SDK. If this is your first time working with the SDK, start with this document.
56XX-PG707-R	*Stacking Guide* This guide describes how to use the discovery and stacking applications provided in this release.

Additionally, starting with this release, please review the RN-SDK65xDNX-R document for DNX Release Notes for SDK 6.5.x. This is a companion guide describing only specific DNX family device changes in this SDK release. Common changes and resolved issues are described within the RN-SDK659-R document. This document is packaged in the release deliverable itself.

Section 3: New devices added to this release

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

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

Section 3.1: Supported XGS Switch Devices

Family	*Devices*	*Description*
None in this release		

Section 3.2: Preview XGS Switch Devices

Family	*Devices*	*Description*
BCM56970	BCM56970 B0	6.4 Tbps Multilayer Switch
BCM88060	BCM88060 A0/B0	Fibre Channel/Fibre Channel-over-Ethernet Mapper
	BCM88068 B0	Quad/Octal Ethernet Port Extender/Buffer
BCM53570	BCM53570 A0	High Port Count Integrated Switch With 100-FX/1G/2.5G/5G/10G/25G-Capable SerDes Lanes
	BCM53575 A0	8x1G + 4x10G
BCM56170	BCM56170 A0	High Port Count Integrated Switch With 100-FX/1G/2.5G/5G/10G/25G-Capable SerDes Lanes
	BCM56172 A0	24xMG-lite + 4x25G + 2x40G/8x10G
	BCM56174 A0	24X1G + 4x10G + 2x40G/8x10G

Section 3.3: Supported External PHYs

Family	*Devices*	*Description*	*Switch qualified against*
None in this release			

Section 3.4: Preview External PHYs

Family	*Devices*	*Description*	*Switch qualified against*
BCM54190	BCM54190 A0/B0	Octal SGMII Copper Gigabit Ethernet Transceiver	BCM56565/BCM56760

Section 4: New Features per Device

Section 4.1: BCM56970 (Tomahawk2) Updates

The Broadcom® BCM56970 family is a class of high performance, high-connectivity network switching devices supporting up to 64 x 100GbE, 128 x 40GbE, 128 x 25GbE, or 128 x 10GbE switch ports. The BCM56970 delivers high-bandwidth, glueless network connectivity up to 6.4 Tbps on a single chip. Overall software is considered GA level quality for the BCM56970 A0 device only since SDK 6.5.8. The device has gone through another release regression cycle and has added further support as listed below.

Section 4.1.1: Device revision support

The BCM56970 B0 revision is supported as Preview in this release.

Section 4.1.2: New features supported on BCM56970

The features described in below table are updated for BCM56970 in this release and will be supported at GA level on B0 silicon.

Table 1: New features supported

Feature	***Note***
PVTMON High Temperature Protection	If any of the 8 PVTMON (0-7) on the chip detects that the current chip temperature is exceeding its pre-programmed threshold value, either of two actions will be taken: ● Generating an interrupt to notify the application software, if PVTMON5 detects the temperature is exceeding its threshold value (default 110C) ● HW logic putting the chip into reset state to protect it from overheating, if any of other 7 PVTMON(0-4, 6-7) detects the temperature is exceeding its threshold value (default 125C)
Resiliency of DLB over ECMP	Provide the resiliency of traffic when a member of DLB ECMP group is deleted or added

Section 4.1.3: Known limitations

- BCM956970 A0 uses the `SC_X4_FINAL_CONFIG_STATUS_FEC` register to report the resolved state for both IEEE FEC and MSA_FEC. The final resolved status register shows "CL91_SINGLE_LANE_BRCM_PROP" when the IEEE FEC is the actual resolved state. This issue is only in reporting the status and functionally the IEEE FEC is resolved correctly.
- For 40G ports, XLPPI interface settings are the same as SR4 as they are similar electrically at the lane level. Hence for use cases of XLPPI, use interface type of SR4 instead.
- Both protocol-based vlan and port-based vlan would fail for the 128th port on Tomahawk2. At maximum 127 ports are supported. For the 128th port, if untagged packet is missed in the previous vlan assignment (e.g. mac based vlan), a default vlan 1 will be assigned. This is also the default behavior of all ports if they are not explicitly vlan assigned. This limitation is being reviewed to be addressed in future releases.

Section 4.2: BCM88060 (Montreal2) B0 Beta support

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

universal port device, the FCoE capability can optionally connect to a host of switches via a standard Ethernet when the FC universal port is needed.

This release supports the following device firmware: BCM8806x version 1.0.9 (rc/firmware/BCM8806x) which should be copied to the SDK boot directory. This firmware binary (BCM8806x) must be present in the SDK boot directory for loading during initialization.

This release provides support for the BCM88060 B0 revision device as well as the BCM88068 B0 SKU.

Section 4.2.1: Status of supported features

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

Table 2: SDK Feature Status

Feature	***Status***	***Note***
Ethernet Repeater mode	Beta	Supported speeds: 10G / 20G / 25G / 40G / 50G / 100G
Ethernet EBE (Extended Buffered for Ethernet mode)	Beta	Supported speeds: 10G / 20G / 25G / 40G / 50G / 100G Supports different speeds for line and system side, but lane counts should be same. Mixed mode support (10G/25G).
Warmboot	Beta	Tested in Linux environment with Ethernet and FCoE.
Ethernet Flexport	Beta	Quad, Tri0, Tri1, dual, single port mode supported.
HiGig	Beta	Supported 42G and 106G
Parallel Download with multiple MT2	Beta	Tested with chained MT2 setup.
FCoE - Fixed mode and A	Beta	Supported speeds: 4G / 8G / 16G / 32G.
FCoE - Flexport	Beta	System side flex supported.
FCoE - Statistics and Events handling	Beta	Tested inband statistics in FCoE setup.
Dynamic switch of Ethernet and FC modes	Beta	Supported.

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

Section 4.2.2: Notes and Known Limitations

- Testing is ongoing with BCM56960, BCM56850, and BCM56768 SVKs on PowerPC with Linux. These combinations are considered Beta in this release.
- Reset of port is required after PRBS tests.
- Retimer and Gearbox mode are not supported for Ethernet.

Section 4.3: BCM56170 (Hurricane3-MG) A0 Beta Support

The Broadcom® BCM56170 and BCM53570 System-on-a-Chip (SoC) series devices feature an integrated high-speed ARM Cortex-A9 processor, embedded Cortex-R5 processor and enterprise-level buffer and table sizes. This release provides Beta support for features that have been validated on BCM56170 A0 silicon.

In addition to BCM56170 A0, BCM56172 A0, BCM56174 A0, BCM53570 A0, and BCM53575 A0 devices are supported at Preview level with the basic sanity tests for the limited core port configurations in this release. The full validation of these devices is subject to be updated in future releases. The following configurations are supported in this release:

- 48Px1G + 4Px25G + 8Px10G
- 16Px1G (SGMII) + 16Px1G (QSGMII) + 4Px25G + 8Px10G
- 32Px1G (QSGMII) + 4Px25G + 8Px10G
- 8Px1G (QGMII) + 4Px10G
- 8Px1G (SGMII) + 4Px10G

Section 4.3.1: Features supported

The following table shows the maturity level of legacy features in this SDK release.

Table 3. Supported Feature Status

Description	***BCM56170 A0 Status***
Legacy Features	
COSQ	Beta
E2ECC	Beta
E2EFC	Beta
Field	Beta
IPMC	GA
KNET	GA
L2	GA
L3	GA
Link	Beta
Mcast	GA
MiM Lite	GA
Mirror	Beta
Multicast	GA
NIV	GA
OAM	Beta
Packet Tx/Rx	Beta
Policer	Beta
Port	Beta
Port Extender	GA
QoS	GA
Rate	Beta
SER	Beta
Stack	Beta
Stat	Beta
STG	GA
Switch Controls	GA
Time/Timesync	Beta
Trunk	GA
uRPF	Beta
VLAN	GA
Warmboot	Preview
WRED	Beta

Section 4.3.2: New and enhanced features supported on BCM56170 A0

The following table shows the maturity level of new and enhanced features in this SDK release.

Table 4. New Feature Status

Description	***BCM56170 A0 Status***	***Notes***
2-Stage Schedulers	Beta	
L2 My Station TCAM	Beta	
L3 VRF	Beta	
RRoCE	Beta	
RTAG7	Beta	
TSN-Preemption	Beta	
TSN-Seamless Redundancy	Beta	Engineering validation completed with traffic tests for for the scenarios of 4 SR boxes over BCM API sequences: **HSR Redbox**, **HSR Quadbox**, **PRP Redbox**, **PRP-HSR interworking box**
TSN-Service Metering	Beta	
TSN-TAS	Beta	
VXLAN Lite	Beta	

Section 4.4: Updated External PHY Support

Section 4.4.1: BCM84888 (Orca) support

The PHY features in this release have been tested with the BCM56860 switch device. In this release we have added Preview support for 5000x mode.

Current device limitations in this release:

1588 is not yet supported.

Section 4.4.2: BCM82332 (Dino) support

In this release, BCM82332 features have been tested with the BCM56860 switch device.

Current device limitations in this release:

100G PT per lane control on line side is not supported

Digital loopback on line side is failing with 1G.

BER on Merlin lanes is not supported
Port detach using API is failing
Inconsistent TX Lane squelch behavior between Falcon and Merlin lane cores

Section 4.4.3: BCM54190 (Seahawks) support

Device is still under active testing in SDK 6.5.9 with BCM56760/BCM56565 switch devices. Beta support for Macsec has been added in this release. Half Duplex 10M is supported at Preview level in this release.

Current device limitations in this release:

Ports may go down during change in speed from 10M AN to 100M Forced when using mdix=ForcedAuto. Setting MDIX mode manually on link partners recovers the link.

Section 4.4.4: Interoperability testing for new PHY/switch combinations

The following switch and PHY combinations have been interoperability tested in the SDK 6.5.9 release. Below lists only the features that have been tested and supported in this release. Please review prior release notes for prior updates made to these combinations.

Section 4.4.4.1: BCM56860 and PHY BCM84328

Features supported:

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

Section 4.4.4.2: BCM56760 and PHY BCM84328

Features supported:

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

Section 4.4.4.3: BCM56760 and PHY BCM84868

Features supported:

10G/5G/2.5G/1G/100M Auto-Negotiation mode supported
Forced Speed mode applicable to 100M only
PCS Loopback
Diagnostics
PRBS (line side), DSC dump, Eyescan and TX Squelch are not supported

System side PRBS and DSC dump supported

Section 4.4.4.4: BCM88670 and PHY BCM82322

Features supported:

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

Section 5: Things to note

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

Section 5.1: PSIRT Reporting Process

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

Section 5.2: Policy on API defaults

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

Section 5.3: Virtual Machine support

In SDK 6.5.8, Broadcom provided support to configure and enable IOMMU/PCI Passthrough running the Broadcom SDK on Intel/AMD based Linux host servers. Additional test coverage has been added in this release.

Section 5.4: SDK releases out of active engineering support

The following releases are out of active engineering support:

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

Customers are recommended to use this release (6.5.9) for new product development or sustaining releases. The following table shows the number of issues and improvements that have been added to our supported SDK releases by device over the past 12 months. While the table shows individual devices, many issues and improvements will apply to multiple products, e.g. BCM56850 and BCM56860, or all XGS products.

Table 5: Resolved issues and improvements per older XGS device family

Device	***Bugs resolved in the last 12 months***	***Improvements added in the past 12 months***
BCM56760/BCM56565 family	21	12
BCM56270 family	23	6
BCM56260 family	105	33
BCM56860 family	217	65
BCM56960 family	172	63
BCM56850 family	131	29
BCM56840_PLUS	32	3
BCM56340 family	27	12
BCM56640 family	86	9
BCM56450 family	82	25
BCM56150 family	4	2
BCM56440 family	18	12
BCM53400 family	16	4

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

Section 5.5: Warmboot Notes and Considerations

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

Section 5.5.1: Validated Warmboot upgrades

Warmboot testing and issue resolution is focused on a majority of recently supported devices including BCM56970 (like to like testing only) and is performed on a limited set of test cases. Warmboot testing is not performed with PHY devices attached. It is recommended that any customer perform their own warmboot testing for their specific environment and use these results as guidance only.

SDK warmboot upgrades in the table below have been validated on specific SVKs in this release.

Software Upgrade	***Validated***
6.4.11 to 6.5.9	Yes
6.5.8 to 6.5.9	Yes

Broadcom resumed performing regular upgrade testing from 6.4.x releases in this release.

Section 5.5.2: Validated Warmboot downgrades

Broadcom has performed limited release downgrade testing from 6.5.9 to 6.5.8 using BCM56960 and BCM56850 family devices. Broadcom is no longer performing downgrade testing from release 6.5.x to release 6.4.x.

Portmod does not support warmboot downgrade on BCM88670, BCM56860, BCM56565, BCM56760, and BCM56970 device families.

Section 5.5.3: Warmboot upgrade/downgrade considerations

- When performing warmboot from 6.5.7 to 6.5.8 or 6.5.8 to 6.5.7 with BCM56160 and BCM53440 family devices using BCM54292 PHYs, a mismatch between MDIX mode values may cause a failure when using API bcm_port_mdix_get after warmboot (PHY-3018).
- On BCM56850 and BCM56860 family devices, when configuring E2ECC message transmission with API bcm_port_congestion_config_set, the block number of the port retrieved was wrong and hence the E2ECC tx was enabled at a wrong place. This has been fixed with jira on SDK 6.5.8. If customer has configured E2ECC message

transmission, SDK-112599 should be patched to the earlier version first, then the warmboot upgrade from old version to 6.5.8 will be successful.

- For devices which support customized mode flex counter feature, customized mode flex port group created by `bcm_stat_group_mode_id_create()` may be deleted unexpectedly after warmboot upgrading from SDK 6.4.x to SDK 6.5.0 or later versions. The issue will not be hit if no customized group mode flex counter is created, or SDK upgrades among SDK 6.5.x.

Section 6: Summary of BCM API changes and enhancements

This section summarizes BCM API changes in this release. Complete documentation will be available in the Network Switching Software Programmer's Guide number 56XX-PG659-R. For the full list of API support by Broadcom device, please reference the file SDK-6.5.x-Support-Matrix.xls in the sdk/RELDOCS directory in the release package.

Section 6.1: Changes in Class of Service Queue Configuration

New features that can be controlled on a gport/cosq basis.

Table: CoSQ Control Type Values

Value	*Description*
bcmCosqControlEgressPriorityOverCast	To allow activating SP before WFQ mode (per OTM port) by: bcm_cosq_control_set(unit, bcmCosqGportTypeLocalPort, 0, bcmCosqControlEgressPriorityOverCast, 1 / 0);

Table: BCM BST CoSQ IDs

flag	*Description*
bcmBstStatIdInvalid	Invalid BST Id and used for sync all the Hardware stats to the Software copy

Table: Type of a resource.

bcmReservationResourceBufferDescriptors	Resource reservation of buffer descriptors.
bcmResourceOcbBuffers	OCB resource in buffers.

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

```
/* Structure holds the key parameters for the object mapping API. */
typedef struct bcm_cosq_obj_map_key_s {
    bcm_gport_t gport;
} bcm_cosq_obj_map_key_t;

/* API for mapping gport (queue) into object_stat_id. */
extern int bcm_cosq_stat_obj_map_set(
    int unit,
    int flags,
    bcm_cosq_obj_map_key_t *key,
    uint32 object_stat_id);

/* Get the object_stat_id mapping for a given gport queue */
extern int bcm_cosq_stat_obj_map_get(
    int unit,
    int flags,
    bcm_cosq_obj_map_key_t *key,
    uint32 *object_stat_id);

/*
 * Structure hold the key parameters for cosq statistics configuration
```

```
 * API.
 */
typedef struct bcm_cosq_stat_info_key_s {
    int command_id;
} bcm_cosq_stat_info_key_t;

/*
 * Structure gather the cosq parameters in order to generate the counter
 * pointer toward the counter processor.
 */
typedef struct bcm_cosq_stat_info_s {
    int enable_mapping;         /* Enable/disable using the object stat id
                                   mapping table when generate the counter
                                   pointer */
    int num_queues_per_entry;   /* Defines how many consecutive VOQs are
                                   Counted in one counter set. */
    uint32 queue_first;        /* Start range of VOQs which will be counted. */
    uint32 queue_last;          /* End range of VOQs which will be counted. */
} bcm_cosq_stat_info_t;


/*
 * API configure cosq statistics parameters per command_id, in order to
 * generate the counter pointer toward the counter processor.
 */
extern int bcm_cosq_stat_info_set(
    int unit,
    int flags,
    bcm_cosq_stat_info_key_t *key,
    bcm_cosq_stat_info_t *config);

/* Get cosq statistics parameters for given command_id */
extern int bcm_cosq_stat_info_get(
    int unit,
    int flags,
    bcm_cosq_stat_info_key_t *key,
    bcm_cosq_stat_info_t *config);

#endif /* BCM_HIDE_DISPATCHABLE */

typedef struct bcm_cosq_static_threshold_s {
    int thresh_set;     /* On threshold. */
    int thresh_clear;   /* Off threshold -- if not hyst, clear = set. */
} bcm_cosq_static_threshold_t;

typedef enum bcm_cosq_ingress_forward_priority_e {
    bcmCosqIngrFwdPriorityNone = -1,
    bcmCosqIngrFwdPriorityLow = 0,
```

```
    bcmCosqIngrFwdPriorityHigh = 1
} bcm_cosq_ingress_forward_priority_t;

typedef enum bcm_cosq_forward_decision_type_e {
    bcmCosqFwdTypeFwd = 0,
    bcmCosqFwdTypeSnoop = 1,
    bcmCosqFwdTypeMirror = 2,
    bcmCosqFwdTypeStatSample = 3
} bcm_cosq_forward_decision_type_t;

typedef uint32 bcm_cosq_threshold_index_t;

/* Threshold Action type */
typedef enum bcm_cosq_threshold_action_type_e {
    bcmCosqThreshActionDrop = 0,    /* Drop threshold */
    bcmCosqThreshActionFc = 0       /* FC threshold */
} bcm_cosq_threshold_action_type_t;

/* Complementary information for static threshold. */
typedef struct bcm_cosq_static_threshold_info_s {
    bcm_gport_t gport;                  /* Target gport. */
    bcm_cosq_threshold_index_t index;
    bcm_cosq_threshold_action_type_t threshold_action;
    bcm_cosq_resource_t resource_type;  /* Resource type. */
    int is_free_resource;               /* If true -  test free resources,
                                           otherwise - test occupied resources.
*/
} bcm_cosq_static_threshold_info_t;


/* Set static threshold parameters. */
extern int bcm_cosq_gport_static_threshold_set(
    int unit,
    uint32 flags,
    bcm_cosq_static_threshold_info_t *thresh_info,
    bcm_cosq_static_threshold_t *threshold);

/* Get static threshold parameters. */
extern int bcm_cosq_gport_static_threshold_get(
    int unit,
    uint32 flags,
    bcm_cosq_static_threshold_info_t *thresh_info,
    bcm_cosq_static_threshold_t *threshold);


/* Index creation macros. */
#define BCM_COSQ_TH_INDEX_DP_SET(dp)        _SHR_COSQ_TH_INDEX_DP_SET(dp)
```

```
#define BCM_COSQ_TH_INDEX_TC_SET(tc)        _SHR_COSQ_TH_INDEX_TC_SET(tc)
#define BCM_COSQ_TH_INDEX_POOL_DP_SET(pool_id, dp)
_SHR_COSQ_TH_INDEX_POOL_DP_SET(pool_id, dp)
#define BCM_COSQ_TH_INDEX_CAST_PRIO_FDWACT_SET(cast, priority, forward_action)
_SHR_COSQ_TH_INDEX_CAST_PRIO_FDWACT_SET(cast, priority, forward_action)

#define BCM_COSQ_GPORT_IS_RCY(gport)  _SHR_COSQ_GPORT_IS_RCY(gport)

#define BCM_COSQ_GPORT_IS_RCY_MIRR(gport)  _SHR_COSQ_GPORT_IS_RCY_MIRR(gport)
```

Section 6.2: Changes in Explicit Congestion Notification (ECN)

Table: ECN Map structure

Parameter	*Description*
action_flags	action flags for ECN. BCM_ECN_MAP_ACTION_FLAGS_f
int_cn	internal cn as key.
inner_ecn	inner packet ip ecn as key
ecn	outer packet ip ecn as key
exp	mpls_exp as key.
new_ecn	new ecn value should be updated in ip packet.
new_exp	new mpls_exp should be updated in MPLS packet.
nonip_action_flags	action flags for ECN. BCM_ECN_MAP_ACTION_FLAGS_f for non-IP packets

Table: ECN DSCP Map Modes

Name	*Purpose*
bcmEcnDscpMapModeDraft	Draft mode
bcmEcnDscpMapModeRfc3168	RFC 3168 mode

API Changes in Explicit Congestion Notification (ECN)

bcm_ecn_map_mode_t_init

To initialize an ecn mapping mode structure.

SYNOPSIS

```
#include <bcm/ecn.h>
bcm_ecn_map_mode_t_init(bcm_ecn_map_mode_t *ecn_map_mode)
```

PARAMETERS

ecn_map_mode	ecn map mode structure.

DESCRIPTION

To initialize an ecn map mode structure.

```
/* ECN map mode structure */
typedef struct bcm_ecn_map_mode_s {
    int dscp;                                   /* -1 to change all entries */
    bcm_ecn_dscp_map_mode_t dscp_map_mode;   /* ecn map mode setting. */
} bcm_ecn_map_mode_t;
```

Valid modes are detailed in table Table 5: ECN DSCP Map Modes .

RETURNS

Nothing

bcm_ecn_map_mode_set

Configure the ECN mapping mode.

SYNOPSIS

```
#include <bcm/ecn.h>
int
bcm_ecn_map_mode_set(
    int unit,
```

```
    bcm_ecn_map_mode_t *ecn_map_mode);
```

PARAMETERS

unit	(IN) Unit number.
ecn_map_mode	(IN) ecn mapping mode information

DESCRIPTION

Configure the ECN mapping mode for a specific DSCP. A dscp value of -1 will cause all the 64 entries to be changed simultaneously.

RETURNS

BCM_E_xxx

---

bcm_ecn_map_mode_get

Retrieve the ECN mapping mode.

SYNOPSIS

```
#include <bcm/ecn.h>
int
bcm_ecn_map_mode_get(
    int unit,
    bcm_ecn_map_mode_t *ecn_map_mode);
```

PARAMETERS

unit	(IN) Unit number.
ecn_map	(IN/OUT) ecn mapping mode information

DESCRIPTION

Retrieve the ECN mapping mode for a specific DSCP.

RETURNS

BCM_E_xxx

Section 6.3: Changes in Fabric

Table: bcm_fabric_threshold_type_t

Source Name	*Queue index is taken from...*
bcmFabricRxWfqDynamicWeightPipeTh	Dynamic weights
bcmFabricMidWfqDynamicWeightPipeTh	
bcmFabricTxWfqDynamicWeightPipeTh	

API Changes in Fabric

bcm_fabric_link_profile_set

The following API associate links with profile (0,1).

SYNOPSIS

```
#include <bcm/fabric.h>
int
bcm_fabric_link_profile_set(
    int unit,
    int profile_id,
    uint32 flags,
    uint32 links_count,
    bcm_port_t links);
```

PARAMETERS

unit	(IN) Unit number.
profile_id	(IN)
flags	(IN)
links_count	(IN)
links	(IN)

RETURNS

int

bcm_fabric_link_profile_get

The following API retrieves the threshold profile for given links.

SYNOPSIS

```
#include <bcm/fabric.h>
int
bcm_fabric_link_profile_get(
    int unit,
    int profile_id,
    uint32 flags,
    uint32 links_count_max,
    uint32 links_count,
    bcm_port_t links);
```

PARAMETERS

unit	(IN) Unit number.
profile_id	(IN)
flags	(IN)
links_count_max	(IN)
links_count	(OUT)
links	(OUT)

RETURNS

int

bcm_fabric_multicast_local_route_set

Enables MC local route for multiple MC groups.

SYNOPSIS

```
#include <bcm/fabric.h>
int
bcm_fabric_multicast_local_route_set(
    int unit,
    uint32 flags,
    uint32 nof_groups,
    bcm_multicast_t groups,
```

```
    int enable);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
nof_groups	(IN) Number of MC groups for which local indication is configured.
groups	(IN) Array of MC groups, of size ‘nof_groups’. Each entry contains the group’s MC-ID.
enable	(IN) Enable/disable as local route.

RETURNS

int

bcm_fabric_multicast_local_route_get

Get MC local route state for multiple MC groups.

SYNOPSIS

```
#include <bcm/fabric.h>
int
bcm_fabric_multicast_local_route_get(
    int unit,
    uint32 flags,
    uint32 nof_groups,
    bcm_multicast_t groups,
    int enable);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
nof_groups	(IN) Number of MC groups for which local indication is configured.
groups	(IN) Array of MC groups, of size ‘nof_groups’. Each entry contains the group’s MC-ID.
enable	(OUT) Enable/disable as local route.

RETURNS

int

bcm_fabric_link_topology_config_set

Configure links topology.

SYNOPSIS

```
#include <bcm/fabric.h>
int
bcm_fabric_link_topology_config_set(
    int unit,
    uint32 flags,
    bcm_module_t destination,
    bcm_fabric_topology_type_t type,
    int links_count,
    bcm_port_t links_array);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
destination	(IN)
type	(IN)
links_count	(IN)
links_array	(IN)

DESCRIPTION

bcm_fabric_topology_type_t

Fabric topology type.

- Load balancing topology.
- Multicast topology.
- All.

RETURNS

int

bcm_fabric_link_topology_config_get

Get the configured links topology.

SYNOPSIS

```
#include <bcm/fabric.h>
int
bcm_fabric_link_topology_config_get(
    int unit,
    uint32 flags,
    bcm_module_t destination,
    bcm_fabric_topology_type_t type,
    int links_count_max,
    int links_count,
    bcm_port_t links_array);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
destination	(IN)
type	(IN)
links_count_max	(IN)
links_count	(OUT)
links_array	(OUT)

RETURNS

int

bcm_fabric_rci_resolution_set

SYNOPSIS

```
#include <bcm/fabric.h>
int
bcm_fabric_rci_resolution_set(
    int unit,
    uint32 flags,
    bcm_fabric_rci_resolution_key_t key,
```

```
    bcm_fabric_rci_resolution_config_t config);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
key	(IN)
config	(IN)

RETURNS

int

```
typedef struct bcm_fabric_rci_resolution_key_s {
    int pipe;
    uint32 shared_rci;
    uint32 guaranteed_rci;
} bcm_fabric_rci_resolution_key_t;


typedef struct bcm_fabric_rci_resolution_config_s {
    uint32 resolved_rci;
} bcm_fabric_rci_resolution_config_t;
```

bcm_fabric_rci_resolution_get

SYNOPSIS

```
#include <bcm/fabric.h>
int
bcm_fabric_rci_resolution_get(
    int unit,
    uint32 flags,
    bcm_fabric_rci_resolution_key_t key,
    bcm_fabric_rci_resolution_config_t config);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
key	(IN)
config	(OUT)

RETURNS

int

Section 6.4: Changes in Field Processor

Table: Data Qualifier Flags

Data qualifier Flag	*Purpose*
BCM_FIELD_DATA_QUALIFIER_WITH_ID	Qualifier ID is provided in qualifier
BCM_FIELD_DATA_QUALIFIER_REPLACE	Replace existing qualifier.
BCM_FIELD_DATA_QUALIFIER_OFFSET_IP4_OPTIONS_ADJUST	Offset does not include IPv4 header options.
BCM_FIELD_DATA_QUALIFIER_OFFSET_IP6_EXTENSIONS_ADJUST	Offset does not include IPv6 extension header.
BCM_FIELD_DATA_QUALIFIER_OFFSET_GRE_OPTIONS_ADJUST	Offset does not include GRE header options.
BCM_FIELD_DATA_QUALIFIER_OFFSET_FLEX_HASH	Flexible Hashing offset qualifier.
BCM_FIELD_DATA_QUALIFIER_OFFSET_NEGATIVE	The 'offset' field is interpreted as negative.
BCM_FIELD_DATA_QUALIFIER_OFFSET_BIT_RES	The 'offset' field is computed with a bit resolution.
BCM_FIELD_DATA_QUALIFIER_OFFSET_PREDEFINED	The 'qualifier' field is used instead of 'offset_base'.
BCM_FIELD_DATA_QUALIFIER_OFFSET_FLEX_HASH	The 'qualifier' field is used to create flex hash entry'.
BCM_FIELD_DATA_QUALIFIER_LENGTH_BIT_RES	The 'length' field is computed with a bit resolution.
BCM_FIELD_DATA_QUALIFIER_STAGE_LOOKUP	To create Data Qualifier for Lookup Stage.
BCM_FIELD_DATA_QUALIFIER_BIT_OFFSET_DNX_ORDER	Control the qualifier bit numbering.

Table: Packet Resolution Values (for bcm_field_qualify_PacketRes)

BCM_FIELD_PKT_RES_xxx	*Purpose*
BCM_FIELD_PKT_RES_L2_ANY	To match any L2 packet. It combines the resolution types BCM_FIELD_PKT_RES_L2BC, BCM_FIELD_PKT_RES_L2UC, BCM_FIELD_PKT_RES_L2UNKNOWN, BCM_FIELD_PKT_RES_L2MCKNOWN and BCM_FIELD_PKT_RES_L2MCUNKNOWN

BCM_FIELD_PKT_RES_L2MC_ANY	To match any L2 MC packet. This options is to combine the resolution types BCM_FIELD_PKT_RES_L2MCKNOW and BCM_FIELD_PKT_RES_L2MCUNKNOWN
BCM_FIELD_PKT_RES_TRILL_ANY	To match any Trill packet. This option is to combine the resolution types BCM_FIELD_PKT_RES_TRILLKNOWN and BCM_FIELD_PKT_RES_TRILLUNKNOWN
BCM_FIELD_PKT_RES_NIV_ANY	To match any NIV packet. This option is to combine resolution types BCM_FIELD_PKT_RES_NIVKNOWN and BCM_FIELD_PKT_RES_NIVUNKNOWN
BCM_FIELD_PKT_RES_MPLS_ANY	To match any FCOE packet. This option is to combine resolution types BCM_FIELD_PKT_RES_MPLSKNOWN, BCM_FIELD_PKT_RES_MPLSL3KNOWN, BCM_FIELD_PKT_RES_MPLSL2KNOWN, BCM_FIELD_PKT_RES_MPLSUNKNOWN and BCM_FIELD_PKT_RES_MPLSMCKNOWN
BCM_FIELD_PKT_RES_MIM_ANY	To match any MIM packet. This option is to combine resolution types BCM_FIELD_PKT_RES_MIMKNOWN and BCM_FIELD_PKT_RES_MIMUNKNOWN
BCM_FIELD_PKT_RES_FCOE_ANY	To match any FCOE packet. This option is to combine resolution types BCM_FIELD_PKT_RES_FCOEKNOWN and BCM_FIELD_PKT_RES_FCOEUNKNOWN
BCM_FIELD_PKT_RES_L3_ANY	To match any L3 packet. This option is to combine resolution types BCM_FIELD_PKT_RES_L3MCKNOWN, BCM_FIELD_PKT_RES_L3MCUNKNOWN, BCM_FIELD_PKT_RES_L3KNOWN and BCM_FIELD_PKT_RES_L3UNKNOWN
BCM_FIELD_PKT_RES_LAST	Must be >= highest valid BCM_FIELD_PKT_RES_*

Table: Field Qualifiers

Qualifier	*Purpose*
FieldQualifyEgressBridgedForwardingDataHigh	Used for ACL-while-bridging. Contains the 64 bits of packet data starting 80 bits after SA

FieldQualifyEgressBridgedForwardingDataLow	Used for ACL-while-bridging. Contains the 56 bits of packet data starting 144 bits after SA
FieldQualifyEgressBridgedAclDataHigh	Used for ACL-while-bridging. Contains the 40 bits of packet data starting 200 bits after SA
FieldQualifyEgressBridgedAclDataLow	Used for ACL-while-bridging. Contains the 32 bits of packet data starting 240 bits after SA
FieldQualifyCount	Always Last. Not a usable value.

Table: Field Forwarding Type

Type	*Purpose*
bcmFieldForwardingTypeCustom1	Custom_1 forwarding
bcmFieldForwardingTypeCustom2	Custom_2 forwarding

Table: Field Qualify OLP header type Format

Formats	*Purpose*
bcmFieldOlpHeaderTypeMacSecEncrypt,	OLP header type for MACSEC Encrypt flow
bcmFieldOlpHeaderTypeMacSecDecrypt,	OLP header type for MACSEC Decrypt flow

Table: Field Actions for bcm_field_action_add

Action	*Description*	*param0*	*param1*
bcmFieldActionMacSecRedirectIpmc	This action redirects multicast MACSEC packets to MACSEC engine and also adds respective OLP header in Egress Pipeline if and only if bcmFieldActionMacSecEncrypt or bcmFieldActionMacSecDecrypt action configured in the entry.	IP Multicast Index	n/a

Table: Group config flags

Group config flag	*Purpose*
BCM_FIELD_GROUP_CREATE_WITH_CYCLE	If set, use created cycle value for field group.

Section 6.5: Changes in Mirroring

Table: BCM Mirror Destination Flags

Name	*Description*
BCM_MIRROR_DEST_OUT_MIRROR_DISABLE	Assert out mirror disable

Section 6.6: Changes in Operations, Administration, And Maintenance

```
/* OAM endpoint object.  This is equivalent to an 802.1ag Maintenance Endpoint
(MEP). */
typedef struct bcm_oam_endpoint_info_s {
...
    bcm_oam_profile_t acc_profile_id; /* Used by accelerated endpoints. */
} bcm_oam_endpoint_info_t;


/* bcm_oam_profile_type_t will represent the oam action profile type. */
typedef enum bcm_oam_profile_type_e {
    bcmOAMProfileIngressLIF = 0,
    bcmOAMProfileEgressLIF = 1,
    bcmOAMProfileIngressAcceleratedEndpoint = 2,
    bcmOAMProfileEgressAcceleratedEndpoint = 3,
    bcmOAMProfileCount = 4
} bcm_oam_profile_type_t;

/* MAC types for OAM packets. Used for action profile configuration. */
typedef enum bcm_oam_dest_mac_type_e {
    bcmOAMDestMacTypeMcast = 0,
    bcmOAMDestMacTypeMyCfmMac = 1,
    bcmOAMDestMacTypeUknownUcast = 2,
    bcmOAMDestMacTypeCount = 3
} bcm_oam_dest_mac_type_t;
```

```
/* Used to identify Loss Measurement type. */
typedef enum bcm_oam_loss_type_e {
    bcmOAMLossTypeSingleEnded = 0,  /* PDU used will be Eth-LM. */
    bcmOAMLossTypeSLM = 1,
    bcmOAMLossTypeDualEnded = 2,    /* PDU used will be CCM. */
    bcmOAMLossTypeCount = 3
} bcm_oam_loss_type_t;


/* Endpoint type used for action profile configuration */
typedef enum bcm_oam_match_type_e {
    bcmOAMMatchTypeMIP = 0,
    bcmOAMMatchTypeMEP = 1,
    bcmOAMMatchTypePassive = 2,
    bcmOAMMatchTypeLowMDL = 3,
    bcmOAMMatchTypeCount = 4
} bcm_oam_match_type_t;

/* OAM action key. */
typedef struct bcm_oam_action_key_s {
    uint32 flags;
    bcm_oam_opcode_t opcode;
    bcm_oam_dest_mac_type_t dest_mac_type;
    bcm_oam_match_type_t endpoint_type;
    uint8 inject;
} bcm_oam_action_key_t;

/* OAM action result. */
typedef struct bcm_oam_action_result_s {
    uint32 flags;
    uint8 count_enable;
    uint8 meter_enable;
    bcm_gport_t destination;
    bcm_oam_loss_type_t lm_type;
    bcm_oam_timestamp_format_t time_format;
} bcm_oam_action_result_t;
```

API Changes in Operations, Administration, And Maintenance

bcm_oam_profile_create

Function allocate oam profile.

SYNOPSIS

```
#include <bcm/oam.h>
int
bcm_oam_profile_create(
    int unit,
    uint32 flags,
    bcm_oam_profile_type_t oam_profile_type,
    bcm_oam_profile_t profile_id);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
oam_profile_type	(IN) Type of profile that user want to allocate.
profile_id	(INOUT) Used as input and output when BCM_OAM_PROFILE_WITH_ID flag is set and used as output for other cases.

RETURNS

int

bcm_oam_profile_delete

Function release oam profile.

SYNOPSIS

```
#include <bcm/oam.h>
int
bcm_oam_profile_delete(
    int unit,
    uint32 flags,
    bcm_oam_profile_t profile_id);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
profile_id	(IN) oam profile id that should be de-allocated.

RETURNS

int

bcm_oam_lif_profile_get

Function return oam lif profile according lif.

SYNOPSIS

```
#include <bcm/oam.h>
int
bcm_oam_lif_profile_get(
    int unit,
    uint32 flags,
    bcm_gport_t lif,
    bcm_oam_profile_t ingress_oam_profile,
    bcm_oam_profile_t egress_oam_profile);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN) Reserved for future use.
lif	(IN) Logical Interface.
ingress_oam_profile	(OUT) Profile of LIF(the value is taken from sw state).
egress_oam_profile	(OUT) Profile of LIF(the value is taken from sw state).

RETURNS

int

bcm_oam_lif_profile_set

Function return oam lif profile according lif. If some end point already exist on given LIF, profile will be updated in OAM-LIF-DB table and stored in sw state.If no endpoint yet defined on given LIF, profile will be stored in sw state.

SYNOPSIS

```
#include <bcm/oam.h>
int
bcm_oam_lif_profile_set(
    int unit,
    uint32 flags,
    bcm_gport_t lif,
    bcm_oam_profile_t ingress_oam_profile,
    bcm_oam_profile_t egress_oam_profile);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN) Reserved for future use.
lif	(IN) Logical Interface.
ingress_oam_profile	(IN) Profile of LIF(the value is taken from sw state).
egress_oam_profile	(IN) Profile of LIF(the value is taken from sw state).

RETURNS

int

bcm_oam_profile_action_get

Function return action.

SYNOPSIS

```
#include <bcm/oam.h>
int
bcm_oam_profile_action_get(
    int unit,
    uint32 flags,
    bcm_oam_profile_t profile_id,
    bcm_oam_action_key_t oam_action_key,
    bcm_oam_action_result_t oam_action_result);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
profile_id	(IN) Oam action profile id
oam_action_key	(IN) Oam action key.

oam_action_result	(OUT) Return oam action result.

RETURNS

int

bcm_oam_profile_action_set

Function return action.

SYNOPSIS

```
#include <bcm/oam.h>
int
bcm_oam_profile_action_set(
    int unit,
    uint32 flags,
    bcm_oam_profile_t profile_id,
    bcm_oam_action_key_t oam_action_key,
    bcm_oam_action_result_t oam_action_result);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
profile_id	(IN) Oam action profile id
oam_action_key	(IN) Oam action key.
oam_action_result	(IN) Return oam action result.

RETURNS

int

Section 6.7: Changes in Policer

```
typedef struct bcm_policer_config_s {
...
    int color_resolve_profile; /* Input data for the resolved table. */
} bcm_policer_config_t;



/* Policer Group Mode Attribute Selectors */
typedef enum bcm_policer_group_mode_attr_e {
...
    bcmPolicerGroupModeAttrOuterVlanCfi = 16, /* Outer VLAN CFI: Possible
Values:<0 or
                                          1> */
    bcmPolicerGroupModeAttrInnerVlanCfi = 17, /* Inner VLAN CFI: Possible
Values:<0 or
                                          1> */
    bcmPolicerGroupModeAttrLast = 18,   /* Last value. Not to be used */
} bcm_policer_group_mode_attr_t;


The bcm_policer_action_t structure enumerates the Global service meter actions

typedef enum bcm_policer_action_e {
...
    bcmPolicerActionGpIntCongestionNotificationNew = 18, /* Replace the
internal congestion
                                             notification value of green packet.
                                             param0: Internal congestion
                                             notification value. */
    bcmPolicerActionYpIntCongestionNotificationNew = 19, /* Replace the
internal congestion
                                             notification value of yellow packet.
                                             param0: Internal congestion
                                             notification value. */
    bcmPolicerActionRpIntCongestionNotificationNew = 20, /* Replace the
internal congestion
                                             notification value of red packet.
                                             param0: Internal congestion
                                             notification value. */
    bcmPolicerActionCount = 21          /* Last value. Not to be used */
 } bcm_policer_action_t;



/* Enum to select the scope of the policer. */
```

```
typedef enum bcm_policer_scope_e {
    bcmPolicerScopeIngressGlobal = 0,
    bcmPolicerScopeIngress = 1,
    bcmPolicerScopeEgress = 2
} bcm_policer_scope_t;

/* Forwarding types enum. */
typedef enum bcm_policer_fwd_types_e {
    bcmPolicerFwdTypeUc = 0,
    bcmPolicerFwdTypeUnkownUc = 1,
    bcmPolicerFwdTypeMc = 2,
    bcmPolicerFwdTypeUnkownMc = 3,
    bcmPolicerFwdTypeBc = 4,
    bcmPolicerFwdTypeMax = 5
} bcm_policer_fwd_types_t;

/*
 * Structure hold the for each forwarding type the offset in the
 * expansion group and if to do meter on it or not.
 */
typedef struct bcm_policer_expansion_group_types_config_s {
    int offset; /* Offset in expansion group */
    int valid;  /* If not valid, packet with this type will be filtered. */
} bcm_policer_expansion_group_types_config_t;

/*
 * Structure holds the policer expansion groups, according the forwarding
 * types.
 */
typedef struct bcm_policer_expansion_group_s {
    bcm_policer_expansion_group_types_config_t config[bcmPolicerFwdTypeMax];
} bcm_policer_expansion_group_t;




/* Enum to select the section in the policer-crps shared memory engines. */
typedef enum bcm_policer_engine_section_e {
    bcmPolicerEngineSectionAll = 0,
    bcmPolicerEngineSectionLow = 1,
    bcmPolicerEngineSectionHigh = 2
} bcm_policer_engine_section_t;

/* Structure holds the policer memory engine key configuration. */
typedef struct bcm_policer_engine_s {
    int core_id;
    int engine_id;                    /* Same units as the crps engines. valid
```

```
                                            engines are the ones that are shared
                                            with crps: 0,1,16,17,18,19,20,21. */
    bcm_policer_engine_section_t section; /* When using dual bucket mode, must
                                            use 'All'. */
} bcm_policer_engine_t;

/*
 * Structure define the database that the engine will attached to and its
 * base stat pointer.
 */
typedef struct bcm_policer_database_attach_config_s {
    int policer_database_handle;    /* Database handle defines the database
                                       that engine belong to. */
    int object_stat_pointer_base;   /* Resolution after expansion. */
} bcm_policer_database_attach_config_t;
```

API Changes in Policer

bcm_policer_expansion_groups_set

API Defines groups of L2 Fwd-Types and decide if to count them and in which offset in the expansion to count them.

SYNOPSIS

```
#include <bcm/policer.h>
int
bcm_policer_expansion_groups_set(
    int unit,
    int flags,
    bcm_core_t core_id,
    bcm_policer_expansion_group_t expansion_group);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
core_id	(IN)
expansion_group	(IN)

RETURNS

int

bcm_policer_expansion_groups_get

API gets the expansion groups of L2 Fwd-Types.

SYNOPSIS

```
#include <bcm/policer.h>
int
bcm_policer_expansion_groups_get(
    int unit,
    int flags,
    bcm_core_t core_id,
    bcm_policer_expansion_group_t expansion_group);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
core_id	(IN)
expansion_group	(OUT)

RETURNS

int

bcm_policer_database_create

API create a policer database (sw create). database represent the meter processor belong to one of the meter commands.It gets as input database handler which gather the key parameters: core, scope, database_id

SYNOPSIS

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

```
bcm_policer_database_create(
    int unit,
    int flags,
    int policer_database_handle,
    bcm_policer_database_config_t config);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
policer_database_handle	(IN) Handle decode the policer database key: core, scope, database_id.
config	(IN)

RETURNS

```
/* Structure holds configuration per policer database. */
typedef struct bcm_policer_database_config_s {
    int expansion_enable; /* Allowed to ingress database 0. if enabled,
expansion size, filter and offsets will be  made according API
bcm_policer_expansion_groups_set. */
    int is_single_bucket; /* Single or dual bucket mode. */
} bcm_policer_database_config_t;
```

bcm_policer_database_destroy

API destroy a policer database

SYNOPSIS

```
#include <bcm/policer.h>
int
bcm_policer_database_destroy(
    int unit,
    int flags,
    int policer_database_handle);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)

policer_database_handle	(IN) Handle decode the policer database key: core, scope, database_id.

RETURNS

int

bcm_policer_control_set

API which contain several policer control types that can be configured.

SYNOPSIS

```
#include <bcm/policer.h>
int
bcm_policer_control_set(
    int unit,
    int flags,
    int policer_database_handle,
    bcm_policer_control_type_t type,
    uint32 arg);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
policer_database_handle	(IN)
type	(IN)
arg	(IN)

DESCRIPTION

bcm_policer_control_type_t

Enum holds the policer control types.

- Filter meter pointers according the dp command. It is a bitmap.
- select the input for the resolve table: policer-profile or dp-command. This type is per ingress/egress only and not per database.
- Select if to count bytes or packets in the internal policer stats. (default=packets).

RETURNS

int

bcm_policer_control_get

API get the value of one of the control types.

SYNOPSIS

```
#include <bcm/policer.h>
int
bcm_policer_control_get(
    int unit,
    int flags,
    int policer_database_handle,
    bcm_policer_control_type_t type,
    uint32 arg);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
policer_database_handle	(IN)
type	(IN)
arg	(OUT)

RETURNS

int

bcm_policer_engine_database_attach

API attach one engine (or half engine) to a policer database.

SYNOPSIS

```
#include <bcm/policer.h>
int
bcm_policer_engine_database_attach(
    int unit,
```

```
    int flags,
    bcm_policer_engine_t engine,
    bcm_policer_database_attach_config_t config);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
engine	(IN)
config	(IN)

RETURNS

int

bcm_policer_engine_database_get

API gets per engine (or half engine), the database it attached to.

SYNOPSIS

```
#include <bcm/policer.h>
int
bcm_policer_engine_database_get(
    int unit,
    int flags,
    bcm_policer_engine_t engine,
    bcm_policer_database_attach_config_t config);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
engine	(OUT)
config	(OUT)

RETURNS

int

bcm_policer_engine_database_detach

API detach one engine (or half engine) from a policer database. From this point, it can be used to crps or other policer database.

SYNOPSIS

```
#include <bcm/policer.h>
int
bcm_policer_engine_database_detach(
    int unit,
    int flags,
    bcm_policer_engine_t engine);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
engine	(OUT)

RETURNS

int

bcm_policer_database_enable_set

API enable/disable policer database.

SYNOPSIS

```
#include <bcm/policer.h>
int
bcm_policer_database_enable_set(
    int unit,
    int flags,
    int policer_database_handle,
    int enable);
```

PARAMETERS

unit	(IN) Unit number.

flags	(IN)
policer_database_handle	(IN)
enable	(IN)

RETURNS

int

bcm_policer_database_enable_get

API gets if policer database is enabled or disabled.

SYNOPSIS

```
#include <bcm/policer.h>
int
bcm_policer_database_enable_get(
    int unit,
    int flags,
    int policer_database_handle,
    int enable);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
policer_database_handle	(IN)
enable	(IN)

RETURNS

int

Section 6.8: Changes in Port Configuration

Table: bcm_port_config_t

Field	*BCM Type*	*Description*
eventor	bcm_pbmp_t	Mask of Eventor ports

control	bcm_pbmp_t	Mask of OLP ports
control	bcm_pbmp_t	Mask of OAMP ports
control	bcm_pbmp_t	Mask of ERP ports

Table: bcm_port_pcs_t

Value	*Description*
bcmPortPCS64b66b15TRsFec	64b66b 15T RsFec
bcmPortPCS64b66b15TLowLatencyRsFec	64b66b 15T LLRSFec

Table: bcm_port_control_t

bcmPortControlLLFCAfterFecEnable	Enable/Disable FEC on LLFC and congestion indication.
bcmPortControlResetSerdes.	Apply reset and release reset to SerDes core. Hard resets PCS and PMD

Table: bcm_port_control_phy_timesync_t

PortControlPhyTimesyncTimestampOffset	the delay from CMIC to the PCS
PortControlPhyTimesyncTimestampAdjust	the adjustment of TS

AM norm modes for aggregated port seeds

Table: bcm_port_phy_timesync_compensation_mode_t

PortPhyTimesyncCompensationModeNone	No compensation
PortPhyTimesyncCompensationModeEarliestLane	Compensate based on the earliest lane
PortPhyTimesyncCompensationModeLatestlane	Compensate based on the latest lane.

PortNifSchedulerLow	Low Priority Scheduler
PortNifSchedulerHigh	High Priority Scheduler
PortNifSchedulerTDM	TDM Priority Scheduler

```
/* Port max priority groups. */
#define BCM_PORT_MAX_NOF_PRIORITY_GROUPS    3

/* Port priority */
#define BCM_PORT_F_PRIORITY_0   1<<1       /* User defined priority 0. */
#define BCM_PORT_F_PRIORITY_1   1<<2       /* User defined priority 1. */
#define BCM_PORT_F_PRIORITY_2   1<<3       /* User defined priority 2. */
#define BCM_PORT_F_PRIORITY_3   1<<4       /* User defined priority 3. */
#define BCM_PORT_F_PRIORITY_TDM 1<<16      /* User defined priority TDM */

/* Priority Scheduler. */
typedef enum bcm_port_nif_scheduler_e {
    bcmPortNifSchedulerLow = 0,     /* Low Priority Scheduler */
    bcmPortNifSchedulerHigh = 1,    /* High Priority Scheduler */
    bcmPortNifSchedulerTDM = 2      /* TDM Priority Scheduler */
} bcm_port_nif_scheduler_t;

/* bcm_port_prio_group_config_t */
typedef struct bcm_port_prio_group_config_s {
    uint32 source_priority;             /* see BCM_PORT_F_PRIORITY_* */
    bcm_port_nif_scheduler_t sch_priority; /* see bcm_port_nif_scheduler_t */
    int num_of_entries;                 /* Indicate how many memory entries
each
                                           priority group takes. Set to -1 for
                                           equal division between all
                                           priorities; otherwise indicate
number
                                           of entries up to 128 per lane */
} bcm_port_prio_group_config_t;

/* bcm_port_prio_config_t */
typedef struct bcm_port_prio_config_s {
    int nof_priority_groups;            /* see BCM_PORT_F_PRIORITY */
    bcm_port_prio_group_config_t
priority_groups[BCM_PORT_MAX_NOF_PRIORITY_GROUPS];
```

```
} bcm_port_prio_config_t;
```

API Changes in Port Configuration

bcm_port_priority_config_set

Configure priority groups for the port. Priority group is meant to connect one or more user priorities (see BCM_PORT_F_PRIORITY_*) to a scheduler priority (High, Low, TDM) Each priority group acts as a logical FIFO. The size of the FIFO can be dynamically specified, in entries.

SYNOPSIS

```
#include <bcm/port.h>
int
bcm_port_priority_config_set(
    int unit,
    bcm_port_t port,
    bcm_port_prio_config_t port_priority_config);
```

PARAMETERS

unit	(IN) Unit number.
port	(IN)
port_priority_config	(IN)

RETURNS

int

bcm_port_priority_config_get

Retrieve the port priority group configuration.

SYNOPSIS

```
#include <bcm/port.h>
int
bcm_port_priority_config_get(
    int unit,
    bcm_port_t port,
    bcm_port_prio_config_t priority_config);
```

PARAMETERS

unit	(IN) Unit number.
port	(IN)
priority_config	(OUT)

RETURNS

int

bcm_port_lane_to_serdes_map_set

Map lanes to serdeses. The API receive all lanes configuration at once. Location in the array represent the lane number (e.g. serdes_map[x] holds the information for lane=x).

SYNOPSIS

```
#include <bcm/port.h>
int
bcm_port_lane_to_serdes_map_set(
    int unit,
    int flags,
    int map_size,
    bcm_port_lane_to_serdes_map_t serdes_map);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
map_size	(IN)
serdes_map	(IN)

RETURNS

int

```
/* This struct holds serdes information (rx and tx) related to a specific lane.
*/
typedef struct bcm_port_lane_to_serdes_map_s {
    int serdes_rx_id;
    int serdes_tx_id;
} bcm_port_lane_to_serdes_map_t;
```

bcm_port_lane_to_serdes_map_get

Map lanes to serdeses. The API receive all lanes configuration at once. Location in the array represent the lane number (e.g. serdes_map[x] holds the information for lane=x).

SYNOPSIS

```
#include <bcm/port.h>
int
bcm_port_lane_to_serdes_map_get(
    int unit,
    int flags,
    int map_size,
    bcm_port_lane_to_serdes_map_t serdes_map);
```

PARAMETERS

unit	(IN) Unit number.
flags	(IN)
map_size	(IN)
serdes_map	(OUT)

RETURNS

int

Section 6.9: Changes in Precision Time Protocol

```
/* PTP TOD Formats */
 typedef enum bcm_ptp_tod_format_e {
     bcmPTPTODFormatString,
     bcmPTPTODFormatUBlox,
     bcmPTPTODFormatChinaTcom,
     bcmPTPTODFormatBCM,
     bcmPTPTODFormatBCMTS,
     bcmPTPTODFormatG8271
 } bcm_ptp_tod_format_t;

/* PTP Synce Clock type enumeration. */
typedef enum bcm_ptp_synce_clock_type_e {
    bcmPTPSynceClockPrimary = 0x00,     /* PTP Synce Clock Primary */
    bcmPTPSynceClockSecondary = 0x01    /* PTP Synce Clock Secondary or Backup
*/
} bcm_ptp_synce_clock_type_t;

/* PTP TOD Channel */
typedef struct bcm_ptp_channel_s {
    bcm_ptp_source_type_t type;         /* PTP Channel Source Type */
    bcm_ptp_timestamp_source_t source;  /* PTP Channel Timestamp Source.
                                           Value depends on type: */
                                        /* if type=SyncETimestamp, value is
                                           port index          */
                                      /* else value is GPIO/TSGPIO Pin index */
    int frequency;                      /* PTP Channel Frequency */
    int tod_index;                      /* PTP Channel Index */
    int freq_priority;                  /* PTP Channel Frequency Priority */
    int freq_enabled;                   /* PTP Channel Frequency Enabled */
    int time_prio;                      /* PTP Channel Time Priority */
    int time_enabled;                   /* PTP Channel Time Enabled */
    int freq_assumed_QL;                /* PTP Channel Frequency Assumed QL */
    int time_assumed_QL;                /* PTP Channel Time Assumed QL */
    int assumed_QL_enabled;             /* PTP Channel Assumed QL Enabled */
    int resolution;                     /* PTP Channel Resolution */
    bcm_ptp_synce_port_t synce_input_port; /* PTP SyncE input port */
    bcm_ptp_synce_clock_type_t synce_input_clk_type; /* PTP SyncE Input Clock
type */
} bcm_ptp_channel_t;

/* Phase Offsets for Path Asymmetry. */
typedef struct bcm_ptp_phase_offset_s {
    int64 reported_phase_offset;   /* Current servo-reported offset between
localtime and the recovered clock */
```

```
    int64 delta_phase_offset;      /* Application-supplied offset to compensate
for known path asymmetry */
    int64 fixed_phase_offset;      /* fixed phase offset */
    int use_fixed_phase_offset;    /* Boolean flag to use fixed_phase_offset or
not */
    int phase_slew_rate_ppb;       /* Max slew rate when supplied phase offset
is changed */
    uint32 sync_phase_offset_ns;   /* Max offset to adjust by slew. Larger
offset will cause phase jump */
} bcm_ptp_phase_offset_t;

typedef struct bcm_tdpll_input_clock_ql_change_cb_data_s {
    int input_clk_index;
    bcm_esmc_quality_level_t prior_ql;
    bcm_esmc_quality_level_t current_ql;
} bcm_tdpll_input_clock_ql_change_cb_data_t;

typedef int (*bcm_tdpll_input_clock_ql_change_cb)
    int unit,
    int stack_id,
    bcm_tdpll_input_clock_ql_change_cb_data_t *cb_data);
```

API Changes in Precision Time Protocol

bcm_tdpll_input_clock_ql_change_callback_register

Register input clock esmc quality change callback.

SYNOPSIS

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

PARAMETERS

unit	(IN) Unit number.
stack_id	(IN) Stack identifier index.
ql_change_cb	(IN) Input clock esmc quality change callback function pointer

DESCRIPTION

Register input clock esmc quality change callback.

RETURNS

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

bcm_tdpll_input_clock_ql_change_callback_unregister

Unregister input clock esmc quality change callback.

SYNOPSIS

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

PARAMETERS

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

DESCRIPTION

Unregister input clock esmc quality change callback.

RETURNS

BCM_E_NONE	Operation completed successfully
BCM_E_xxx	Operation failed

Section 6.10: Changes in Packet Transmit and Receive

Table: bcm_rx_cfg_t Structure Changes

Member	*Type*	*Description*
pkt_size	int	Maximum bytes to support in a packet
pkts_per_chain	int	See above; number of packets to be associated with a chain. This setting affects the through-put and memory demands of RX. In general, more packets per chain results in better throughput at the expense of more memory use.
global_pps	int	Global packets-per-second rate limit; 0 disables this limit. This setting limits the number of RX interrupts per second and may improve overall system performance by reducing the CPU resources required by RX operation. A reasonable limit will also prevent a misconfigured system from 'freezing up' due to excessive RX work.
max_burst	int	Maximum number of packets to allow in without any rate limiting. If undesired packet loss occurs with rate limiting enabled, try increasing this value before increasing the global rate limit.
chan_cfg	bcm_rx_chan_cfg_t[BCM_CMICX_RX_CHANNELS]	Per-channel configuration. See (unknown XREF bcm_rx_chan_cfg_t) .
rx_alloc	(unknown XREF bcm_rx_alloc_f)	Packet data allocation routine
rx_free	(unknown XREF bcm_rx_free_f)	Packet data deallocation routine
num_of_cpu_addresses	int	Number of cpu addresses to be configured

cpu_address	int*	The cpu addresses

```
/* Max. number of CMICx RX channels. */
#define BCM_CMICX_RX_CHANNELS   8
```

Section 6.11: Changes in Statistics

```
typedef enum bcm_stat_val_e {
...
    /*** Additional Broadcom stats ***/
+snmpBcmRxRsFecBitError,               /* Fabric RX RS-FEC bit error counter.
(Broadcom-specific). */
   snmpBcmRxRsFecSymbolError,            /* RX RS-FEC symbol error counter.
(Broadcom-specific). */
...
} bcm_stat_val_t;

 /* LIF Counting Source. */
 typedef struct bcm_stat_lif_counting_source_s {
    bcm_stat_counter_source_type_t type;
    int command_id;
    int stif_counter_id;
    uint32 offset;
    int engine_range_offset;            /* Positive or negative offset to align
                                           the profile range to the engine
                                           range. relevant only for out lif
                                           counting in QAX/QUX. */
 } bcm_stat_lif_counting_source_t;
```

API Changes in Statistics

bcm_stat_engine_t_init

init bcm_stat_engine_t

SYNOPSIS

```
#include <bcm/stat.h>
int
bcm_stat_engine_t_init(
    bcm_stat_engine_t stat_engine);
```

PARAMETERS

stat_engine	(INOUT)

RETURNS

void

bcm_stat_counter_explicit_input_data_t_init

init bcm_stat_counter_explicit_input_data_t

SYNOPSIS

```
#include <bcm/stat.h>
int
bcm_stat_counter_explicit_input_data_t_init(
    bcm_stat_counter_explicit_input_data_t stat_counter_explicit_input_data);
```

PARAMETERS

stat_counter_explicit_input_data	(INOUT)

RETURNS

void

bcm_stat_eviction_t_init

init bcm_stat_eviction_t

SYNOPSIS

```
#include <bcm/stat.h>
int
bcm_stat_eviction_t_init(
    bcm_stat_eviction_t stat_eviction);
```

PARAMETERS

stat_eviction	(INOUT)

RETURNS

void

bcm_stat_counter_set_map_t_init

init bcm_stat_counter_set_map_t

SYNOPSIS

```
#include <bcm/stat.h>
int
bcm_stat_counter_set_map_t_init(
    bcm_stat_counter_set_map_t stat_counter_set_map);
```

PARAMETERS

stat_counter_set_map (INOUT)

RETURNS

void

---

bcm_stat_expansion_data_mapping_t_init

init bcm_stat_expansion_data_mapping_t

SYNOPSIS

```
#include <bcm/stat.h>
int
bcm_stat_expansion_data_mapping_t_init(
    bcm_stat_expansion_data_mapping_t stat_expansion_data_mapping);
```

PARAMETERS

stat_expansion_data_mapping (INOUT)

RETURNS

void

---

bcm_stat_counter_interface_key_t_init

init bcm_stat_counter_interface_key_t

SYNOPSIS

```
#include <bcm/stat.h>
int
bcm_stat_counter_interface_key_t_init(
```

```
    bcm_stat_counter_interface_key_t stat_counter_interface_key);
```

PARAMETERS

stat_counter_interface_key	(INOUT)

RETURNS

void

bcm_stat_expansion_select_t_init

init bcm_stat_expansion_select_t

SYNOPSIS

```
#include <bcm/stat.h>
int
bcm_stat_expansion_select_t_init(
    bcm_stat_expansion_select_t stat_expansion_select);
```

PARAMETERS

stat_expansion_select	(INOUT)

RETURNS

void

bcm_stat_counter_interface_t_init

init bcm_stat_counter_interface_t

SYNOPSIS

```
#include <bcm/stat.h>
int
bcm_stat_counter_interface_t_init(
    bcm_stat_counter_interface_t stat_counter_interface);
```

PARAMETERS

stat_counter_interface	(INOUT)

RETURNS

void

bcm_stat_engine_enable_t_init

init bcm_stat_engine_enable_t

SYNOPSIS

```
#include <bcm/stat.h>
int
bcm_stat_engine_enable_t_init(
    bcm_stat_engine_enable_t stat_engine_enable);
```

PARAMETERS

stat_engine_enable	(INOUT)

RETURNS

void

bcm_stat_counter_output_data_t_init

init bcm_stat_counter_output_data_t

SYNOPSIS

```
#include <bcm/stat.h>
int
bcm_stat_counter_output_data_t_init(
    bcm_stat_counter_output_data_t stat_counter_output_data);
```

PARAMETERS

stat_counter_output_data	(INOUT)

RETURNS

void

Section 6.12: Changes in Trunking (Link Aggregation)

```
#define BCM_TRUNK_PSC_C_LAG             18         /* Use Consistent Lag tables
                                                      to choose members. */
#define BCM_TRUNK_PSC_WEIGHTED_PROFILE  19         /* Use Weighted profile to
                                                      choose members. */

#define BCM_TRUNK_FLAG_LOCAL                0x0800     /* If set, trunk can
                                                          contain only local
                                                          members. */
#define BCM_TRUNK_FLAG_DONT_ALLOCATE_PP_PORTS 0x1000     /* If set, trunk will
not
                                                          allocate pp ports by
                                                          default. */

#define BCM_TRUNK_MEMBER_MEMBER_DISABLE     0x0100     /* Member will not
                                                          receive traffic as
                                                          result of hashing */

/* Trunk profile information structure. */
typedef struct bcm_trunk_psc_profile_info_s {
    int weight_array; /* Array of weights for members in trunk. */
    int weight_array_size; /* actual Number of members in weight array */
```

```
    uint32 psc_flags; /* BCM_TRUNK_PSC_xxx */
    int max_nof_members_in_profile; /* max number of members in a profile */
} bcm_trunk_psc_profile_info_t;

 /* Trunk group attributes structure. */
 typedef struct bcm_trunk_info_s {
     uint32 flags; /* BCM_TRUNK_FLAG_xxx. */
    int psc; /* Port selection criteria. */
    bcm_trunk_psc_profile_info_t psc_info; /* Port selection criteria
information. */
     int ipmc_psc; /* Port selection criteria for software IPMC trunk
resolution. */
     int dlf_index; /* DLF/broadcast port for trunk group. */
     int mc_index; /* Multicast port for trunk group. */
     int ipmc_index; /* IPMC port for trunk group. */
     int dynamic_size; /* Number of flows for dynamic load balancing. Valid
values are 512, 1k, doubling up to 32k */
     int dynamic_age; /* Inactivity duration, in microseconds. */
     int dynamic_load_exponent; /* The exponent used in the exponentially
weighted moving average calculation of historical member load. */
     int dynamic_expected_load_exponent; /* The exponent used in the
exponentially weighted moving average calculation of historical expected member
load. */
 } bcm_trunk_info_t;

/* Trunk ID information structure. */
typedef struct bcm_trunk_id_info_s {
    int pool_index;     /* pool index of trunk_id */
    int group_index;    /* group index of trunk_id */
} bcm_trunk_id_info_t;

/* Trunk Local attributes structure. */
typedef struct bcm_trunk_local_attributes_s {
    uint32 core_bitmap; /* bitmap for the cores on which the local trunk
members
                           can exist */
    int nof_priorities; /* number of port priorities of the trunk */
} bcm_trunk_local_attributes_t;
```

API Changes in Trunking (Link Aggregation)

bcm_trunk_find

Get the trunk group ID for a given system port, specified by Module ID and Port number

SYNOPSIS

```
#include <bcm/trunk.h>
int
bcm_trunk_find(
    int unit,
    bcm_module_t modid,
    bcm_gport_t gport,
    bcm_trunk_t tid);
```

PARAMETERS

unit	BCM device number
modid	Module ID
port	Port ID.
tid	(OUT) Trunk ID

DESCRIPTION

Retrieve the trunk group ID for a given system port, specified by its Module ID and Port number.

RETURNS

BCM_E_NONE	Success
BCM_E_UNAVAIL	Feature not supported by the SoC hardware
BCM_E_INIT	Trunk module has not been initialized
BCM_E_BADID	Invalid or unsupported Module ID or Port number
BCM_E_NOT_FOUND	Trunk ID not found
BCM_E_XXX	Other possible errors; for details, see (unknown XREF error_codes)

bcm_trunk_id_set

API used to set trunk_ID according to trunk ID info

SYNOPSIS

```
#include <bcm/trunk.h>
int bcm_trunk_id_set(int unit,  bcm_trunk_id_info_t *id_info, bcm_trunk_t *tid)
```

PARAMETERS

unit	BCM device number
id_info	
tid	

DESCRIPTION

API used to set trunk_ID according to trunk ID info

RETURNS

BCM_E_NONE	No Error
BCM_E_UNAVAIL	Feature unavailable
BCM_E_PARAM	Invalid Parameter
BCM_E_XXX	Error occurred

bcm_trunk_local_attributes_set

API used to set trunk's local attributes

SYNOPSIS

```
#include <bcm/trunk.h>
int bcm_trunk_local_attributes_set(int unit, bcm_trunk_t tid, uint32 flags,
bcm_trunk_local_attributes_t *local_attributes)
```

PARAMETERS

unit	BCM device number
tid	
flags	
local_attributes	

DESCRIPTION

API used to set trunk's local attributes

RETURNS

BCM_E_NONE	No Error
BCM_E_UNAVAIL	Feature unavailable
BCM_E_PARAM	Invalid Parameter
BCM_E_XXX	Error occurred

bcm_trunk_local_attributes_get

API used to get trunk's local attributes

SYNOPSIS

```
#include <bcm/trunk.h>
int bcm_trunk_local_attributes_get(int unit, bcm_trunk_t tid, uint32 flags,
bcm_trunk_local_attributes_t *local_attributes)
```

PARAMETERS

unit	BCM device number
tid	
flags	
local_attributes	

DESCRIPTION

API used to get trunk's local attributes

RETURNS

BCM_E_NONE	No Error
BCM_E_UNAVAIL	Feature unavailable
BCM_E_PARAM	Invalid Parameter
BCM_E_XXX	Error occurred

bcm_trunk_local_attributes_t_init

init bcm_trunk_local_attributes_t

SYNOPSIS

```
#include <bcm/trunk.h>
void
bcm_trunk_local_attributes_t_init(
    bcm_trunk_local_attributes_t local_attributes);
```

PARAMETERS

local_attributes (INOUT)

RETURNS

void

Section 6.13: Changes in BCM types

```
#define BCM_IF_INVALID          -1

/* Types of casts. */
typedef enum bcm_cast_e {
    bcmCastUC = 0,  /* Unicast */
    bcmCastMC = 1,  /* Multicast */
    bcmCastAll = -1 /* both Unicast and Multicast */
} bcm_cast_t;


/*
 * Set and get macros which decode the database key parameters and store
 * it in database handle.
 */
#define BCM_POLICER_DATABASE_HANDLE_SET(database_handle, is_ingress, is_global,
core_id, database_id)  _SHR_POLICER_DATABASE_HANDLE_SET(database_handle,
is_ingress, is_global, core_id, database_id)
#define BCM_POLICER_DATABASE_HANDLE_IS_INGRESS_GET(database_handle)
_SHR_POLICER_DATABASE_HANDLE_IS_INGRESS_GET(database_handle)
#define BCM_POLICER_DATABASE_IS_GLOBAL_GET(database_handle)
_SHR_POLICER_DATABASE_HANDLE_IS_GLOBAL_GET(database_handle)
#define BCM_POLICER_DATABASE_CORE_ID_GET(database_handle)
```

```
_SHR_POLICER_DATABASE_HANDLE_CORE_ID_GET(database_handle)
#define BCM_POLICER_DATABASE_DATABASE_ID_GET(database_handle)
_SHR_POLICER_DATABASE_HANDLE_DATABASE_ID_GET(database_handle)


/*
 * Set and get macros which decode the database handle and the meter id
 * (called offset) in policer_id.
 */
#define BCM_POLICER_ID_SET(policer_id, database_handle, offset)
_SHR_POLICER_ID_SET(policer_id, database_handle, offset)
#define BCM_POLICER_ID_OFFSET_GET(policer_id)
_SHR_POLICER_ID_METER_INDEX_GET(policer_id)
#define BCM_POLICER_ID_DATABASE_HANDLE_GET(policer_id)
_SHR_POLICER_ID_DATABASE_HANDLE_GET(policer_id)
```

Section 6.14: Changes in VLAN Management

```
/* Configure VLAN Translation entry with dummy VP */
#define BCM_VLAN_ACTION_SET_DUMMY_VP        0x00000400
```

Section 6.15: Changes in VXLAN Management

```
#define BCM_VXLAN_VNI_INVALID   -1         /* Invalid VXLAN VNI */
```

Section 7: Test Statistics

Section 7.1: How to read the data

In cases where tables are shown below, the tables represent a spread of data gathered per device, per suite, and per release. The percentages represent the aggregate rate of failure for that suite when run against all variants of the family of devices. This data does not include results from DNX device regressions.

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

Section 7.2: Overview

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

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

Section 7.2.1: Linux kernel versions used in this release

In SDK 6.5.9, the following Linux kernel versions were used in our development and regression cycles with these external CPUs:

- BCM958625XMC (RSX): 3.6.5
- BCM9COMX2XMC86 (XLR): 4.4.6
- BCM958712D4XMC (SLK): 3.14.29

Please refer to the Broadcom Network Switching Software Platform Guide for more details about these CPUs. In this release, BCM958712D4XMC (SLK) was used in limited testing while BCM9XLP208XMC (WRX) and BCM98548PPCXMC (GTO) are no longer part of our main focus for regression activity.

Testing using ARM-based CPU subsystem integrated processors (IPROC) was performed using kernel 4.4.39.

Section 7.3: Total Tests

The data below represents the number of unique cases for each release. The goal is to increase test coverage release over release but there may be instances where tests are consolidated which may yield a net reduction from one version to the next. Note that although a particular test case will execute for each and every chip, it is only counted once.

	sdk-6.5.9	***sdk-6.5.8***	***sdk-6.5.7***
golden	153	153	153
warmboot	4348	4320	3248
auth	17	17	17
bfd	117	116	112
bhh	159	159	150
chip	9	9	9
coe	579	579	579
cosq	807	778	773
custom	7	7	7
ea	108	108	108
eav	19	19	19

extender	49	49	49
fabric	7	7	7
failover	10	10	10
fcoe	37	37	37
field	1725	1725	1686
higigproxy	129	129	129
infra	114	114	114
ipfix	17	17	17
ipmc	114	114	114
l2	344	344	344
l2gre	33	33	33
l3	562	556	553
l3.alpm	550	550	550
link	26	26	26
mim	46	46	46
mirror	175	174	173
misc	21	21	21
mpls	541	541	541
multicast	29	29	29
niv	66	66	66
oam	400	400	397

pkt	44	44	44
port	484	485	466
proxy	38	38	38
ptp	115	115	115
qos	62	62	13
rate	21	21	21
rtag7	59	58	50
rx	25	25	25
ser	245	245	157
stack	117	117	117
stat	438	438	439
stg	42	42	42
switch	217	217	217
time	33	33	33
tlvMsg	13	13	13
trill	47	47	47
trunk	252	252	252
tunnel	133	133	133
subport	31	31	31
vlan	245	242	242
vxlan	219	219	219

wlan	17	17	17
Test Suite Total	14215	14147	12848

Section 7.4: API Test Results

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

Section 7.5: PHY Test Results

Starting with SDK 6.5.3, we have increased our focus and coverage on testing specific PHY and switch combinations in order to improve our quality in this area. The tables below represent specific results from our interoperability and regression testing for the release. We have been continually working to improve our results and coverage in ongoing releases.

Section 7.5.1: SQA External PHY

Switch Device	***External Phy Device***	***Total Tests***	***% Fail***
56960_A0	phy82381_100G	173	1.16%
56960_A0	phy82764_10G	173	2.89%
56860_A0	phy82780_10G	37	0.00%
56860_A0	phy82780_40G	37	0.00%
56860_A0	phy82792_100G	173	2.31%
56867_A1	phy82764_10G	173	0.58%
56867_A1	phy82764_40G	173	0.58%

56960_B0	phy82864_100G	173	2.31%
56960_B0	phy82864_100G_alt	173	4.05%
56960_B0	phy82864_40GPt	173	2.89%
56960_B0	phy82864_40GPt_alt	173	4.62%
56960_B0	phy82864_10G	173	6.36%
56960_B0	phy82864_10G_alt	173	6.94%
56960_B0	phy82864_11G	173	2.89%
56960_B0	phy82864_11G_alt	173	3.47%
56960_B0	phy82864_42GPt	173	1.16%
56960_B0	phy82864_42GPt_alt	173	2.89%
56960_B0	phy82864_106G	173	1.73%
56960_B0	phy82864_106G_alt	173	2.89%
56960_B0	phy82864_40G2x20	173	6.36%
56960_B0	phy82864_40G2x20_alt	173	6.94%
56960_B0	phy82864_42G2x20	173	2.89%
56960_B0	phy82864_42G2x20_alt	173	3.47%
56960_B0	phy82864_40GMux	173	6.36%
56960_B0	phy82864_50G	173	2.31%
56960_B0	phy82864_25G	173	4.05%
56867_A1	phy82332_100GPt	26	0.00%
56867_A1	phy82332_100G_gbox	26	7.69%
56867_A1	phy82332_106GPt	26	0.00%

56867_A1	phy82332_40G	26	7.69%
56867_A1	phy82332_42G	26	7.69%
56867_A1	phy82332_10G	26	15.38%
56867_A1	phy82332_11G	26	15.38%
56860_A1	phy82332_retimer_1G	26	7.69%
56860_A1	phy82332_retimer_10G	26	23.08%
56860_A1	phy82332_retimer_11G	26	30.77%
56860_A1	phy82332_retimer_40G	26	15.38%
56860_A1	phy82332_retimer_100G	26	0.00%
56765_A0	phy54140_1G_Copper	173	0.58%
56765_A0	phy54140_1G_Fiber	173	0.58%
56565_A0	phy84868_10G	173	1.16%
56565_A0	phy84868_10G	173	1.16%

Section 7.5.2: Interop External PHY

P2P Suite

Switch Device	*Port Macro*	*Total Tests*	*% Fail*
56560_B0	HG_G40_84328_42G	60	13.33%
56560_B0	XE_G40_84328_10G	122	4.92%
56560_B0	XE_G40_84328_1G	21	14.29%
56560_B0	XE_G40_84328_40G	122	4.92%
56560_B0	XE_MT_84757_10G	60	0.00%
56760_A0	XE_MT_84744_40G	60	0.00%

56760_A0	XE_MT_84744_10G	84	5.95%
56860_A1	XE_QD28_82780_40G	141	0.00%
56860_A1	XE_QD28_82780_10G	246	0.81%
56860_A1	XE_SESTO_82764_40G_MUX	135	0.00%
56860_A1	CE_SESTO_82792_100G_343	204	0.00%
56860_A1	XE_SESTO_82764_40G_PT	174	0.00%
56860_A1	XE_SESTO_82764_10G_PT	162	0.00%
56860_A0	XE_DINO_82332_40G_AN	152	0.66%
56860_A0	CE_DINO_82332_100G_PT_L1_10	242	0.83%
56860_A0	HG_DINO_82332_106G_GB	228	0.00%
56860_A0	CE_DINO_82332_100G_PT	242	0.83%
56860_A0	CE_DINO_82332_100G_GB_L2_11	307	3.58%
56860_A0	HG_DINO_82332_11G	109	0.92%
56860_A0	HG_DINO_82332_42G	127	1.57%
56860_A0	CE_DINO_82332_100G_PT_L2_11	242	0.83%
56860_A0	HG_DINO_82332_106G_PT	246	0.00%
56860_A0	HG_DINO_82332_10G	63	0.00%
56860_A0	HG_DINO_82332_106G_GB_L0_9	186	0.00%
56860_A0	CE_DINO_82332_100G_GB_L1_10	307	3.58%
56860_A0	XE_DINO_82332_10G	383	1.31%
56860_A0	CE_DINO_82332_100G_GB	339	2.06%
56860_A0	XE_DINO_82332_40G	187	3.21%
56860_A0	HG_DINO_82332_11G_RETIMER	125	1.60%
56860_A0	CE_DINO_82332_100G_RETIMER	242	0.83%
56860_A0	XE_DINO_82332_10G_RETIMER	194	2.58%
56860_A0	GE_DINO_82332_1G_RETIMER	168	4.76%
56860_A0	XE_DINO_82332_40G_RETIMER	191	2.62%
56860_A1	XE_G40_84328_10G	148	6.08%
56860_A1	XE_G40_84328_40G	122	4.92%

56860_A0	XE_G28_82322_10G	122	0.00%
56860_A0	XE_G28_82322_40G	122	1.64%
56860_A0	XE_G28_82322_1G	28	0.00%
56860_A0	XE_54210	32	25.00%

Loopback Suite

Switch Device	*Port Macro*	*Total Tests*	*% Fail*
56560_B0	HG_G40_84328_42G	10	0.00%
56560_B0	XE_G40_84328_10G	8	0.00%
56560_B0	XE_G40_84328_1G	2	0.00%
56560_B0	XE_G40_84328_40G	8	0.00%
56560_B0	XE_MT_84757_10G	8	0.00%
56760_A0	XE_MT_84744_40G	8	12.50%
56860_A1	XE_QD28_82780_40G	33	0.00%
56860_A1	XE_QD28_82780_10G	77	0.00%
56860_A1	XE_SESTO_82764_40G_MUX	25	0.00%
56860_A1	CE_SESTO_82792_100G_343	47	0.00%
56860_A1	XE_SESTO_82764_40G_PT	30	0.00%
56860_A1	XE_SESTO_82764_10G_PT	30	0.00%
56860_A0	XE_DINO_82332_40G_AN	5	0.00%
56860_A0	CE_DINO_82332_100G_PT_L1_10	56	0.00%
56860_A0	HG_DINO_82332_106G_GB	56	0.00%
56860_A0	CE_DINO_82332_100G_PT	56	0.00%
56860_A0	CE_DINO_82332_100G_GB_L2_11	63	0.00%
56860_A0	HG_DINO_82332_11G	15	0.00%

56860_A0	HG_DINO_82332_42G	25	0.00%
56860_A0	CE_DINO_82332_100G_PT_L2_11	56	0.00%
56860_A0	HG_DINO_82332_106G_PT	56	0.00%
56860_A0	HG_DINO_82332_10G	10	0.00%
56860_A0	HG_DINO_82332_106G_GB_L0_9	31	0.00%
56860_A0	CE_DINO_82332_100G_GB_L1_10	63	0.00%
56860_A0	XE_DINO_82332_10G	75	18.67%
56860_A0	CE_DINO_82332_100G_GB	63	0.00%
56860_A0	XE_DINO_82332_40G	35	0.00%
56860_A0	HG_DINO_82332_11G_RETIMER	30	0.00%
56860_A0	CE_DINO_82332_100G_RETIMER	56	0.00%
56860_A0	XE_DINO_82332_10G_RETIMER	40	0.00%
56860_A0	GE_DINO_82332_1G_RETIMER	40	60.00%
56860_A0	XE_DINO_82332_40G_RETIMER	35	0.00%
56860_A1	XE_G40_84328_10G	40	7.50%
56860_A1	XE_G40_84328_40G	32	3.13%
56860_A0	XE_G28_82322_10G	8	0.00%
56860_A0	XE_G28_82322_40G	8	0.00%
56860_A0	XE_G28_82322_1G	2	0.00%
56860_A0	XE_54210	8	0.00%
56960_B0	HG_MADURA_82864_42G_PT	66	0.00%
56960_B0	CE_MADURA_82864_100G_ALT	88	0.00%
56960_B0	XE_MADURA_82864_10G	66	0.00%
56960_B0	HG_MADURA_82864_42G_DUAL_ALT	66	0.00%
56960_B0	XE_MADURA_82864_40G_PT_ALT	66	0.00%
56960_B0	XE_MADURA_82864_25G	22	27.27%
56960_B0	HG_MADURA_82864_42G_PT_ALT	66	0.00%
56960_B0	XE_MADURA_82864_50G	22	18.18%
56960_B0	XE_MADURA_82864_40G_MUX	66	36.36%

56960_B0	HG_MADURA_82864_11G	66	0.00%
56960_B0	HG_MADURA_82864_106G_ALT	88	1.14%
56960_B0	XE_MADURA_82864_50G_ALT	22	13.64%
56960_B0	HG_MADURA_82864_42G_MUX	66	43.94%
56960_B0	XE_MADURA_82864_40G_DUAL	66	24.24%
56960_B0	HG_MADURA_82864_42G_DUAL	66	25.76%
56960_B0	XE_MADURA_82864_25G_ALT	22	4.55%
56960_B0	HG_MADURA_82864_11G_ALT	66	0.00%
56960_B0	XE_MADURA_82864_10G_ALT	66	0.00%
56960_B0	XE_MADURA_82864_40G_DUAL_ALT	66	0.00%
56960_B0	HG_MADURA_82864_106G	88	43.18%
56960_B0	CE_MADURA_82864_100G	88	32.95%
56960_B0	XE_MADURA_82864_40G_PT	66	0.00%
56960_B1	CE_MADURA_82864_100G_RETIMER_ULL	77	48.05%
56960_B1	XE_MADURA_82864_25G_ALT_RETIMER	22	0.00%
56960_B1	XE_MADURA_82864_40G_DUAL_ALT_RETIMER	66	36.36%
56960_B1	XE_MADURA_82864_40G_PT_ALT_RETIMER	66	0.00%
56960_B1	HG_MADURA_82864_42G_PT_ALT_RETIMER	66	0.00%
56960_B1	XE_MADURA_82864_10G_RETIMER	66	0.00%
56960_B1	XE_MADURA_82864_50G_RETIMER	22	9.09%
56960_B1	HG_MADURA_82864_11G_ALT_RETIMER	66	0.00%
56960_B1	HG_MADURA_82864_106G_RETIMER	88	30.68%
56960_B1	CE_MADURA_82864_100G_ALT_RETIMER	88	2.27%
56960_B1	HG_MADURA_82864_11G_RETIMER	66	0.00%
56960_B1	XE_MADURA_82864_50G_ALT_RETIMER	22	0.00%
56960_B1	XE_MADURA_82864_25G_RETIMER	22	4.55%
56960_B1	HG_MADURA_82864_42G_PT_RETIMER	66	0.00%

56960_B1	HG_MADURA_82864_42G_DUAL_ALT_RETIMER	66	0.00%
56960_B1	XE_MADURA_82864_40G_MUX_RETIMER	66	36.36%
56960_B1	XE_MADURA_82864_40G_DUAL_RETIMER	66	24.24%
56960_B1	CE_MADURA_82864_100G_RETIMER	88	19.32%
56960_B1	XE_MADURA_82864_10G_ALT_RETIMER	66	0.00%
56960_B1	HG_MADURA_82864_42G_DUAL_RETIMER	66	24.24%
56960_B1	XE_MADURA_82864_40G_PT_RETIMER_ULL	55	0.00%
56960_B1	XE_MADURA_82864_40G_PT_RETIMER	66	0.00%

Section 7.5.3: Interop Internal PHY

P2P Suite

Switch Device	*Port Macro*	*Total Tests*	*% Fail*
56960_B1	HG_64XD53	190	5.79%
56960_B1	HG_32X106	124	9.68%
56960_B1	HG_128X27	116	11.21%
56960_B1	XE_32X100	450	4.00%
56960_B1	XE_64XD50	513	2.92%
56960_B1	XE_128X25	411	3.65%
56860_A1	HG_32X42	108	0.00%
56860_A1	XE_32X40	248	8.06%
56860_A1	HG_8x100_343	31	12.90%
56860_A1	CE_8X100_343_IEEE	139	11.51%
56860_A1	HG_104x10	36	0.00%

56860_A1	XE_104x10	234	0.00%
56860_A1	CE_8x100_442	144	12.50%
56260_A0	GE_VIPER_12x2P5	36	0.00%
56260_A0	EAGLE_XE_4x10	36	0.00%
56260_A0	GE_VIPER_24x1	39	0.00%
56260_A0	XE_VIPER_6x10	30	0.00%
56260_A0	EAGLE_GE_4x1	12	0.00%
56160_A0	SGMII_TSCE	144	0.00%
56160_A0	QSGMII_TSCE	144	1.39%
56160_A0	QSGMII_QTC	108	0.00%
56160_A0	SGMII_QTC	126	0.00%
56160_A0	SGMII_GPHY	108	0.00%
56160_A0	QSGMII_GPHY	108	0.00%
56565_A0	XE_MAKO_1G	12	0.00%
56565_A0	XE_MAKO_10G	16	0.00%
56565_A0	XE_MAKO_5G	4	0.00%
56565_A0	XE_MAKO_RTMR	4	0.00%
56765_A0	XE_18X40	204	2.94%
56765_A0	HG_TSCF_X2	552	2.36%
56765_A0	HG_TSCF_X4	456	2.41%
56765_A0	XE_TSCE_X1	164	6.71%
56765_A0	HG_18X42	108	0.00%
56765_A0	XE_TSCF_X2	359	16.16%
56765_A0	HG_72X11	108	0.00%
56765_A0	XE_TSCF_X4	363	0.55%
56765_A0	XE_TSCE_X2	139	11.51%
56765_A0	HG_TSCE_X1	96	0.00%
56765_A0	HG_16X27	192	10.94%
56765_A0	XE_72X10	117	0.00%

56765_A0	HG_TSCE_X2	96	0.00%
56765_A0	XE_16X25	275	8.00%

Loopback Suite

Switch Device	*Port Macro*	*Total Tests*	*% Fail*
56960_B1	HG_64XD53	66	0.00%
56960_B1	HG_32X106	44	0.00%
56960_B1	HG_128X27	55	20.00%
56960_B1	XE_32X100	88	4.55%
56960_B1	XE_64XD50	99	2.02%
56960_B1	XE_128X25	77	2.60%
56860_A1	HG_32X42	22	54.55%
56860_A1	XE_32X40	66	54.55%
56860_A1	HG_8x100_343	11	54.55%
56860_A1	CE_8X100_343_IEEE	44	54.55%
56860_A1	HG_104x10	11	54.55%
56860_A1	XE_104x10	66	54.55%
56860_A1	CE_8x100_442	44	54.55%
56260_A0	GE_VIPER_12x2P5	28	0.00%
56260_A0	EAGLE_XE_4x10	22	0.00%
56260_A0	GE_VIPER_24x1	30	0.00%
56260_A0	XE_VIPER_6x10	10	0.00%
56260_A0	EAGLE_GE_4x1	10	0.00%
56160_A0	SGMII_TSCE	22	0.00%
56160_A0	QSGMII_TSCE	22	0.00%

56160_A0	QSGMII_QTC	33	0.00%
56160_A0	SGMII_QTC	44	0.00%
56160_A0	SGMII_GPHY	33	0.00%
56160_A0	QSGMII_GPHY	33	3.03%
56565_A0	XE_MAKO_1G	8	0.00%
56765_A0	XE_18X40	44	4.55%
56765_A0	HG_TSCF_X2	60	0.00%
56765_A0	HG_TSCF_X4	40	0.00%
56765_A0	XE_TSCE_X1	77	71.43%
56765_A0	HG_18X42	22	0.00%
56765_A0	XE_TSCF_X2	45	22.22%
56765_A0	HG_72X11	33	0.00%
56765_A0	XE_TSCF_X4	35	0.00%
56765_A0	XE_TSCE_X2	33	0.00%
56765_A0	HG_TSCE_X1	22	0.00%
56765_A0	HG_16X27	20	0.00%
56765_A0	XE_72X10	22	0.00%
56765_A0	HG_TSCE_X2	22	0.00%
56765_A0	XE_16X25	30	16.67%

Section 7.6: Static Code Analysis

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

Section 7.6.1: Unresolved Static Code Analysis Issues

Area	***Open Issues SDK 6.5.9***	***Open Issues SDK 6.5.8***	***Open Issues SDK 6.5.7***	***Open Issues SDK 6.5.6***	***Open Issues SDK 6.5.5***	***Open Issues SDK 6.5.4***	***New baseline as of 3/1/16***	***Open Issues SDK 6.5.3***	***Open Issues SDK 6.5.2***
DNX	8	11	47	63	124	107	120	104	61
XGS	6	25	10	15	23	37	90	86	16
SerDes	10	12	10	38	25	23	35	34	12
Common	8	3	4	27	32	42	45	42	11
Total	32	51	71	143	204	209	330	300	101

Section 8: Service Impacting Defects

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

Table 5: Resolved Service Impacting Defects

Reference	*Chips*	*Affected Versions*	*Errata Synopsis*	*Details*
SDK-120543	BCM5645x, BCM5626x, BCM5646x, BCM5627x families	SDK releases prior to 6.5.9	Performing the flex-port operation on any port will reset the MMU settings on all ports in the device.	SDK was reconfiguring the MMU settings for all ports while performing the flex-port operation on any given port instead of just updating the MMU settings only on the ports involved in flex-port operation. As a result the MMU/COSQ controls configured on ports, not involved in flex-port operation, was getting reset to default values. After the flex-port operation, it could lead to unexpected traffic behavior on ports not participating in flex-port operation.
SDK-119478	88470_A0, 88660_A0, 88670_A0	6.5.9	If destroy API call should fail, it may not return an error.	The return value of various *_destroy_all() APIs was not checked. As a result individual *_destroy() APIs may not have resumed successfully. Bug fixes in *_destroy() APIs were fixed separately as applicable.
SDK-117997	BCM5686x family	SDK releases prior to 6.5.9	Under certain initial configuration, when flexing ports from 100G to 40G, MMU buffers were stuck and getting exhausted.	Under certain mixed 40G/100G configuration, the slot distribution of TDM calendar didn’t satisfy the jitter/spacing requirements on some 40G ports. As a result, MMU TDM scheduler failed to allocate enough bandwidth to these 40G ports within certain window

				period, which caused the MMU buffer exhaustion.
SDK-106616	BCM5664x, BCM5685x, BCM5686x families	6.4.11, 6.5.2, 6.5.3, 6.5.4, 6.5.5, 6.5.6, 6.5.7, 6.5.8	Memory lock issue may be seen between port enable and switch scheduling operation.	When bcm_port_enable_set and bcm_cosq_gport_sched_set are called, a mutex lock problem may be encountered.

Section 9: Resolved Issues for 6.5.9

Starting with SDK 6.5.9, Broadcom now provides a list of resolved improvements and defects in a sortable spreadsheet format rather than a static table in this document. For the full resolved list, please reference the file SDK-6.5.x-Resolved-Issues.xlsx in the RELDOCS directory in the release package.

Section 10: Unresolved Issues for 6.5.9

The following open Urgent priority issues remain unresolved in SDK 6.5.9. These are in process of being evaluated for inclusion in a future SDK release:

Number	***CSP***	***Chips***	***Errata For 6.5.9***
PHY-1980	979443	56340_A0	MDIX control at 1G and 100M
PHY-2102	999972	56334_A0 56334_B0	Lane 0 is not synchronized between BCM84728 and enduro(BCM56334_B0)
PHY-2168	1011776	84846_A0	84846 copper medium can not work when fiber prefer and 8475x is connected to 84846 line side XFI
PHY-2383	1003096	56860_A0 56860_A1	Enzo: BCM56860 with GearBox BCM84793 port attach fails in 10G
PHY-2503	863828	56850_A0 56850_A1 56850_A2	BCM84784: Since upgrading 6.3.8 last 2 ports have errors when DAC is inserted.
PHY-2525	1057318	56640_A0 56640_A1 56640_B0	Errors when initializing BCM84846 in phy8481.c
PHY-2863		56340_A0	10G SFP port flapped on a Helix 4 based customer platform (with ext PHY) when the peer interface status was brought from "down" to "up"

PHY-2902	1123900	84756_A0 84756_C0	SDK-6.5.7 fails to download firmware to PHY 84756
PHY-3027	1132741	56860_A0 56860_A1	BCM84858 EEE is not working
SDK-119861	1137717	56960_A0 56960_B0 56960_B1	TH CMIC block is not generating interrupts and SDK init fails.
SDK-120878	1140594	56860_A0 56860_A1	1G SFP in 10G SFP+ port is failing in few scenarios with SDK6.4.10
SDK-120916	1141674	56960_A0 56960_B0 56960_B1	IFP crash and IFP memory leak issues
SDK-118159		88670_A0 88680_A0	Add SW WA for PVT-mon issue - phymod API

Section 11: Device and Platform Support

For the full list of Broadcom switch and PHY devices supported in the SDK, please reference the file SDK-6.5.9-Device-Matrix.xlsx in the RELDOCS directory in the release package.

Section 12: Compatibility

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

The following table shows new feature support added in Firmware releases for switch devices compatible with the corresponding SDK release. Please refer to the appropriate Network Switching SDK Firmware release notes publication (56XX0_88XX0_FW-RNxxx-R) for the indicated version below for full details. Note there was no new device support added in SDK 6.5.8.

	SDK-6.5.9	SDK-6.5.7	SDK-6.5.6	SDK-6.5.5	SDK-6.5.4	SDK-6.5.3
4.3.4	BCM88270					
4.3.3		BCM56760 BCM56565	BCM56270 BCM56760 BCM56565			
4.3.2				BCM56270 BCM56965 BCM88470		
4.3.1					BCM56760 BCM56560 BCM56565	
4.3.0						BCM56760 BCM56560 BCM56565

Section 12.2: BMACSEC SDK Compatibility Matrix

Switch SDK Release	***BMACSEC Release***
6.5.0	4.14
6.5.1	4.14
6.5.2	4.15

6.5.3	4.15
6.5.4	4.15
6.5.5	4.15
6.5.6	4.16
6.5.7	4.16
6.5.8	4.17
6.5.9	4.18

Section 12.3: iMACSEC SDK Compatibility Matrix

Starting in SDK 6.5.6, Broadcom introduced a new macsec software package called iMACSec. This is specifically for use with the BCM54190 integrated PHY driver.

Switch SDK Release	*iMACSEC Release*
6.5.6	1.0
6.5.7	1.0
6.5.8	1.1
6.5.9	1.1

Section 12.4: PHY Firmware Compatibility Matrix

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

PHY Core	**6.5.4 Firmware Versions**	**6.5.5 Firmware Versions**	**6.5.6 Firmware Versions**	**6.5.7 Firmware Versions**	**6.5.8 Firmware Versions**	**6.5.9 Firmware Versions**
BCM84861	01.00.00	01.00.00	01.00.00	01.00.00	01.00.00	01.00.00

BCM84864	01.00.00	01.00.00	01.00.00	01.00.00	01.00.00	01.00.00
BCM84848	1.09.01	1.09.01	1.09.01	1.09.01	1.09.01	1.09.01
BCM84888	N/A	N/A	00.00.08	00.00.10	00.00.10	A0: 01-00-03 B0: 20.00.03
BCM84868	01.00.00	01.00.00	01.00.00	01.00.00	01.00.00	01.00.00
BCM84858	01.03.02	01.03.02	01.03.02	01.03.02	01.03.04	01.03.04
BCM84856	01.03.02	01.03.02	01.03.02	01.03.02	01.03.04	01.03.04
BCM84744	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0130 (B0/C0)	0x0130 (B0/C0)	0x0130 (B0/C0)	0x0130 (B0/C0)
BCM84757	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)
BCM84328	R029	R029	R029	R029	R029	R029
BCM82322/28 B1	0xF	0xF	0xF	0xF	0xF	0xF
BCM82780	0x23	0x23	0x23	0x28	0x28	0x28
BCM82752	0x23	0x23	0x23	0x28	0x28	0x28
BCM82758	0x23	0x23	0x23	0x28	0x28	0x28
BCM82764	D017	D017	D017	D019	D019	D019
BCM82790	D017	D017	D017	D019	D019	D019
BCM82792	D017	D017	D017	D019	D019	D019
BCM82796	D017	D017	D017	D019	D019	D019
BCM82381	D013	D013	D013	D015	D015	D015

BCM82209	D013	D013	D013	D015	D015	D015
BCM82073	D013	D013	D013	D015	D015	D015
BCM82864	D00C	D00C	D00C	D00E	D00E	D00E
BCM82332	D006	D008	D008	D009	D009	D009
Eagle	D10F_0D	D10F_0D	D10F_13	D10F_13	D10F_13	D10F_13
Falcon	D10B_0C	D10B_0E	D10B_13	D10B_13	D10B_13	D10B_14

Section 13: SDK Externally Licensed Software Components

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

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

SFlow	http://www.inmon.com/technology/sflowlicense.txt	N/A - see Section 13.8

Section 13.1: EDITLINE License terms and conditions

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

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

Added files:
        sysvxworks.c Makefile

Changed functionality:
        Merged unix.h into editline.h
        M-P and M-N now behave like tcsh.
        list_history(count) routine displays history
        Commented out completion
        Changed rl_complete and rl_list_possib into caller-settable
                global functions
        Don't ring bell on TAB if word is already complete

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

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

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

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

  ------------------------------------------------------------------------
```

```
$Revision: 1.7 $

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

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

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

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

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

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

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

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

Section 13.2: LIBXML2 - XML C parser terms and conditions

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

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

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

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

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

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

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

Copyright

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

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

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

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

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

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


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

Section 13.3: ED Editor License terms and conditions

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

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

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

Section 13.4: CINT parser license terms and conditions

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

Removed files:

None

Added files:

None

Changed functionality:

None

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

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

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

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

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

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

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

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

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

Section 13.5: BIGDIGITS license terms and conditions

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

David Ireland and DI Management Services Pty Limited make no
representations concerning either the merchantability of this software
or the suitability of this software for any particular purpose. It is
```

```
provided "as is" without express or implied warranty of any kind. Our
liability will be limited exclusively to the refund of the money you
paid us for the software, namely nothing. By using the software you
expressly agree to such a waiver. If you do not agree to the terms, do
not use the software.
```

Section 13.6: APIMODE parser license terms and conditions

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

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

Section 13.7: Wind River Systems license terms and conditions

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

Section 13.8: SFlow license terms and conditions

Broadcom provides several API modules that refer to SFlow by name, specifically Field, Mirror, Port, and Switch. All are implemented as per IETF RFC-3176. Please review the separate sflowlicense.txt file for terms of the agreement used by Broadcom in our implementation.

Section 14: Feedback

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